



DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(51) Classification internationale des brevets ⁷ : H04L 29/06	A1	(11) Numéro de publication internationale: WO 00/56030 (43) Date de publication internationale: 21 septembre 2000 (21.09.00)
--	-----------	--

(21) Numéro de la demande internationale: PCT/FR00/00625
(22) Date de dépôt international: 15 mars 2000 (15.03.00)
(30) Données relatives à la priorité:
99/03172 15 mars 1999 (15.03.99) FR
(71) Déposant (pour tous les Etats désignés sauf US): BULL CP8 [FR/FR]; 68, route de Versailles, Boîte postale 45, F-78430 Louveciennes (FR).
(72) Inventeur; et
(75) Inventeur/Déposant (US seulement): URIEN, Pascal [FR/FR]; 4, rue du Ruisseau St Prix, F-78450 Villepreux (FR).
(74) Mandataire: BULL S.A.; Corlu, Bernard, PC58D20, 68, route de Versailles, F-78434 Louveciennes Cedex (FR).

(81) Etats désignés: AU, CA, CN, JP, KR, SG, US, brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Publiée

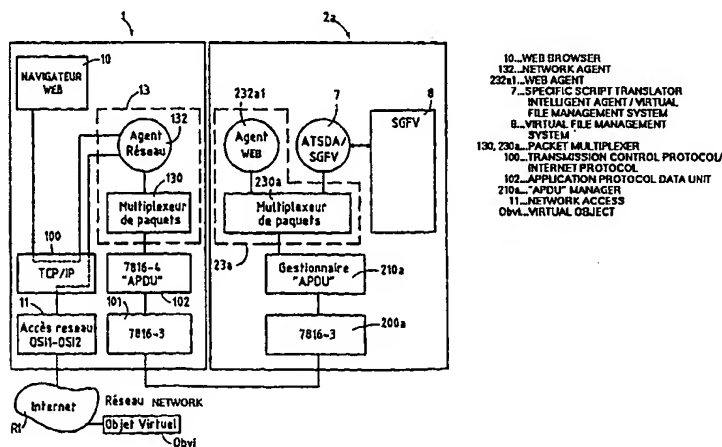
*Avec rapport de recherche internationale.
Avant l'expiration du délai prévu pour la modification des revendications, sera republiée si des modifications sont reçues.*

(54) Title: SYSTEM FOR ACCESSING AN OBJECT USING A "WEB" BROWSER CO-OPERATING WITH A SMART CARD

(54) Titre: SYSTEME D'ACCES A UN OBJET A L'AIDE D'UN NAVIGATEUR DE TYPE "WEB" COOPERANT AVEC UNE CARTE A PUCE

(57) Abstract

The invention concerns an architecture for securely accessing virtual objects (*Obvi*) distributed in systems connected to the Internet (*Ri*), and for obtaining an instance thereof. Said access is performed, via a smart card (2a), through a "WEB" browser (10). The terminal (1) and the smart card (2a) comprise each a specific protocol layer (13, 23a). The latter contains intelligent agents (132, 232a₁) for setting up two-way data exchange sessions, thereby enabling the smart card (2a) to have a "WEB" server functionality. The smart card (2a) also comprises intelligent agents, called script translators, and a virtual file management system (8) co-operating with a specialised script translator intelligent agent (7). Each virtual object (*Obvi*) is associated with a virtual file of the virtual file management system (8). The specialised intelligent agent (7) presents to the browser (10) the list of accessible virtual objects (*Obvi*) and generates methods for accessing said objects.



(57) Abrégé

L'invention concerne un procédé et une architecture permettant d'accéder, de façon sécurisée, à des objets virtuels (*Obv_i*) répartis dans des systèmes connectés au réseau Internet (*RI*), et d'en obtenir une instance. Cet accès s'effectue, via une carte à puce (2a), par l'intermédiaire d'un navigateur de type "WEB" (10). Le terminal (1) et la carte à puce (2a) comprennent chacun une couche protocolaire spécifique (13, 23a). Cette dernière comprend des agents intelligents (132, 232a₁) pour l'établissement de sessions d'échanges bidirectionnels de données, ce qui permet à la carte à puce (2a) de présenter une fonctionnalité de serveur "WEB". La carte à puce (2a) comprend aussi des agents intelligents, dits traducteurs de scripts, et un système de gestion de fichier virtuel (8) coopérant avec un agent intelligent traducteur de scripts spécialisé (7). Chaque objet virtuel (*Obv_i*) est associé à un fichier virtuel du système de gestion de fichier virtuel (8). L'agent intelligent spécialisé (7) présente au navigateur (10) la liste des objets virtuels accessibles (*Obv_i*) et génère des méthodes d'accès à ces objets.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	ES	Espagne	LS	Lesotho	SI	Slovénie
AM	Arménie	FI	Finlande	LT	Lituanie	SK	Slovaquie
AT	Autriche	FR	France	LU	Luxembourg	SN	Sénégal
AU	Australie	GA	Gabon	LV	Lettonie	SZ	Swaziland
AZ	Azerbaïdjan	GB	Royaume-Uni	MC	Monaco	TD	Tchad
BA	Bosnie-Herzégovine	GE	Géorgie	MD	République de Moldova	TG	Togo
BB	Barbade	GH	Ghana	MG	Madagascar	TJ	Tadjikistan
BE	Belgique	GN	Guinée	MK	Ex-République yougoslave de Macédoine	TM	Turkménistan
BF	Burkina Faso	GR	Grèce	ML	Mali	TR	Turquie
BG	Bulgarie	HU	Hongrie	MN	Mongolie	TT	Trinité-et-Tobago
BJ	Bénin	IE	Irlande	MR	Mauritanie	UA	Ukraine
BR	Brésil	IL	Israël	MW	Malawi	UG	Ouganda
BY	Bélarus	IS	Islande	MX	Mexique	US	Etats-Unis d'Amérique
CA	Canada	IT	Italie	NE	Niger	UZ	Ouzbékistan
CF	République centrafricaine	JP	Japon	NL	Pays-Bas	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norvège	YU	Yougoslavie
CH	Suisse	KG	Kirghizistan	NZ	Nouvelle-Zélande	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	République populaire démocratique de Corée	PL	Pologne		
CM	Cameroon	KR	République de Corée	PT	Portugal		
CN	Chine	KZ	Kazakhstan	RO	Roumanie		
CU	Cuba	LC	Sainte-Lucie	RU	Fédération de Russie		
CZ	République tchèque	LI	Liechtenstein	SD	Soudan		
DE	Allemagne	LK	Sri Lanka	SE	Suède		
DK	Danemark	LR	Libéria	SG	Singapour		
EE	Estonie						

SYSTEME D'ACCES A UN OBJET A L'AIDE D'UN NAVIGATEUR DE TYPE "WEB" COOPERANT AVEC UNE CARTE A PUCE

L'invention concerne un système embarqué contenant des informations permettant d'instancier un objet situé sur un réseau, et un procédé pour instancier cet objet.

- 5 L'invention concerne plus particulièrement encore un procédé d'accès sécurisé à cet objet.

Dans le cadre de l'invention, le terme "objet" doit être considéré dans son sens le plus général. Il englobe de nombreux types de ressources informatiques, tels que des fichiers textes, des fichiers images ou des fichiers multimédias (vidéo, son, etc.). Il englobe également des transactions ou des connexions à un système informatique, selon un protocole donné.

Dans le premier cas, on parlera ci-après d'objets statiques, car leur instance ne dépend pas du temps. Dans le second cas, on parlera d'objets dynamiques, car leur instance varie en fonction du temps. On peut citer, à titre d'exemple non limitatif, dans le cadre d'un réseau de type Internet, une connexion de type "Telnet".

Toujours dans le cadre de l'invention, le terme "station d'utilisateur" doit être compris dans un sens général. La station d'utilisateur précitée peut être notamment constituée par un ordinateur personnel fonctionnant sous divers systèmes d'exploitation, tels WINDOWS ou UNIX (tous deux étant des marques déposées). Elle peut être aussi constituée par une station de travail, un ordinateur portable ou un terminal de carte, dit dédié.

De même, dans le cadre de l'invention, le terme "réseau " englobe tout réseau comportant un ensemble de serveurs reliés entre eux, notamment un réseau planétaire dans lequel l'information est transportée de bout en bout. Il s'agit notamment du réseau Internet, de tout réseau dans lequel les échanges de données s'effectuent selon un protocole du type internet, des réseaux privés d'entreprises ou similaires, dits "intranet", et des réseaux les prolongeant vers

l'extérieur, dits "extranet". Il peut notamment s'agir aussi d'un réseau GSM (de l'anglais Global System Mobile), ATM, UMTS, GPRS (de l'anglais Global Packet Radio System), d'un réseau dit « Wireless Network » comme par exemple I3E 802.11, BLUE TOOTH.

- 5 Dans ce qui suit, sans en limiter en quoi que ce soit la portée, on se placera dans le cadre de l'application préférée de l'invention, sauf mention contraire. On considérera donc une station d'utilisateur, que l'on appellera simplement "terminal", munie d'un lecteur de carte à puce et connecté à un réseau de type Internet.
- 10 Un système d'application à base de carte à puce comporte généralement les éléments principaux suivants :
- une carte à puce ;
 - un système hôte constituant le terminal précité ;
 - un réseau de communication, à savoir le réseau Internet dans l'application
- 15 préférée ;
- et un serveur d'application connecté au réseau.

La figure 1A illustre schématiquement un exemple d'architecture de ce type. Le terminal 1, par exemple un ordinateur individuel, comporte un lecteur 3 de carte à puce 2. Ce lecteur 3 peut être ou non physiquement intégré dans le terminal

- 20 1. La carte à puce 2 comporte un circuit intégré 20 dont des connexions d'entrées-sorties affleurent en surface de son support pour autoriser une alimentation en énergie électrique et des communications avec le terminal 1. Ce dernier comprend des circuits d'accès à un réseau de transmissions de données *R/I*. Ces circuits dépendent, notamment, de la nature du réseau *R/I* et
- 25 du terminal 1. A titre d'exemple, il peut s'agir d'une carte réseau pour un réseau de type local ou d'un modem pour se connecter à une ligne téléphonique commutée ou à un réseau numérique à intégration de services ("RNIS"), pour se connecter au réseau Internet, par exemple via un prestataire de services Internet ("Internet Service Provider" ou "ISP", selon la terminologie anglo-
- 30 saxonne).

Le terminal 1 comprend naturellement tous les circuits et organes nécessaires à son bon fonctionnement, et qui n'ont pas été représentés dans un but de simplification du dessin : unité centrale, mémoires vive et fixe, mémoire de masse à disque magnétique, lecteur de disquette et/ou de CédéRom, etc.

- 5 Habituellement, le terminal 1 est aussi relié à des périphériques classiques, intégrés ou non, tels un écran de visualisation 5 et un clavier 6.

- Le terminal 1 peut être mis en communication avec des serveurs ou tous systèmes informatiques connectés au réseau *RI*, dont un seul, 4, est illustré sur la figure 1A. Par « serveur », on entend tout serveur d'information
- 10 apte à traiter des protocoles de communication, soit pour donner accès à des documents, soit pour donner accès à des machines. Dans le cas de l'application préférée de l'invention, les circuits d'accès 11 mettent le terminal 1 en communication avec les serveurs 4 grâce à un logiciel particulier 10, appelé navigateur de type "WEB", ou "browser" selon la terminologie anglo-saxonne.
- 15 Celui-ci permet d'accéder à diverses applications réparties sur l'ensemble du réseau *RI*, généralement selon un mode "client-serveur". Par « navigateur », on entend tout moyen offrant les fonctions suivantes :

- visualisation d'une page, notamment d'une page au standard « SGML » (de l'anglais « Standard Generalized Markup Language », ou langage
- 20 de balisage standard généralisé) ;

-rapatriement de ressources offertes dans la page.

- Cette fonction navigateur correspond à celle visée par le terme anglais « browser ». Une page SGML contient des attributs de présentation, et des liens vers d'autres documents SGML, ou « hyper-liens » vers le monde
- 25 extérieur, c'est-à-dire encore des URI (de l'anglais Unified Resource Identifier, ou Identifiant universel de ressources).

Quant au langage SGML, il comporte de façon connue en soi plusieurs dialectes, dont HTML, XML, et WML.

Habituellement, les communications sur les réseaux s'effectuent conformément à des protocoles répondant à des standards comprenant plusieurs couches logicielles superposées. Dans le cas d'un réseau *R/* de type Internet, les communications s'effectuent selon des protocoles spécifiques à ce type de communications, qui seront détaillés ci-après, mais qui comprennent également plusieurs couches logicielles. Le protocole de communication est choisi en fonction de l'application plus particulièrement visée : interrogation de pages "WEB", transferts de fichiers, courrier électronique (e-mel, ou "e-mail" selon la terminologie anglo-saxonne), forums ou nouvelles ("news" selon la terminologie anglo-saxonne), etc.

L'architecture logique du système comprenant un terminal, un lecteur de carte à puce et la carte à puce, est représentée schématiquement par la figure 1B. Elle est décrite par la norme ISO 7816, qui elle-même comportent plusieurs sous-ensembles :

- 15 - ISO 7816-1 et 7816-2, en ce qui concerne les dimensions et le marquage des cartes ;
- ISO 7816-3, en ce qui concerne le transfert de données entre le terminal et la carte à puce ; et
- ISO 7816-4, en ce qui concerne la structure du jeu d'ordres et le format des commandes.

Sur la figure 1B, du côté terminal 1, on n'a représenté que les couches répondant à la norme ISO 7816-3, référencées 101, et le gestionnaire d'ordres "APDU" (norme ISO 7816-4), référencé 102. Du côté carte à puce 2, les couches répondant à la norme ISO 7816-3 sont référencées 200 et le gestionnaire d'ordres "ADPU" (norme ISO 7816-4) est référencé 201. Les applications sont référencées $A_1, \dots, A_i, \dots, A_n$; n étant le nombre maximum d'applications présentes sur la carte à puce 2.

Une application "cardlet" (marque déposée), A_i , présente dans la carte à puce 2 (figure 1A), dialogue avec le terminal 1 au moyen d'un jeu d'ordres. Ce jeu présente typiquement des ordres d'écriture et des ordres de lecture. Le format des ordres est connu sous l'abréviation anglo-saxonne de "APDU" (pour

"Application Protocol Data Unit"). Il est défini par la norme ISO 7816-4 précitée. Un "APDU" de commande est noté "*APDU.command*" et un "APDU" de réponse est noté "*APDU.response*". Les "APDU" sont échangés entre le lecteur de carte et la carte à puce au moyen d'un protocole spécifié par la norme ISO 7816-3 précitée (par exemple en mode caractère : T=0 ; ou en mode bloc : T=1).

Lorsque la carte à puce 2 inclut plusieurs applications distinctes, comme illustré sur la figure 1B, on parle de carte multi-applicative. Cependant, le terminal 1 dialogue avec une seule application à la fois. Une application A_i se présente, par exemple, sous la forme d'une pièce de logicielle, dite "applet", en langage "JAVA" (marque déposée), que l'on appellera ci-après "cardlet". La sélection d'un "cardlet" particulier A_i est obtenu à l'aide d'un "APDU" du type sélection ("SELECT"). Une fois ce choix effectué, les "APDU" qui le suivent sont acheminés vers ce "cardlet". Un "APDU SELECT" nouveau a pour effet d'abandonner l'application en cours et d'en choisir une autre. Le sous-ensemble logiciel gestionnaire des "APDU" 201 permet de choisir une application particulière A_i dans la carte à puce 2, de mémoriser l'application ainsi choisie, et de transmettre et/ou recevoir des "APDU" vers et depuis cette application.

En résumé de ce qui vient d'être décrit, la sélection d'une application A_i et le dialogue avec celle-ci s'effectue par échanges d'ordres "APDU". On suppose que les applications A_i sont des applications conventionnelles, que l'on appellera ci-après "GCA" (pour "Generic Card Application" ou application de carte générique).

Dans un système d'applications à base de carte à puce, comme illustré par l'architecture de la figure 1B, cette dernière peut se voir dévolu diverses fonctions, notamment des fonctions de sécurité. Il est en effet avantageux de stocker les données liées à la sécurité (mots de passe, droits d'accès, etc.) dans une carte à puce qui peut être conservée par l'utilisateur. En outre les données étant enregistrées dans une mémoire fixe, sous une forme qui peut

être chiffrée, elles ne sont pas facilement modifiables, ni même directement lisibles de l'extérieur.

Cependant, il est à noter que la carte 3 ne peut communiquer directement avec les navigateurs du commerce, sauf à modifier le code de ces derniers.

5 Les cartes à puce actuelles, qui par ailleurs sont conformes aux standards rappelés ci-dessus, ont une configuration matérielle et logicielle qui ne permet pas non plus de communiquer directement avec le réseau Internet. En particulier, elles ne peuvent recevoir et transmettre des paquets de données, selon l'un ou l'autre des protocoles utilisés sur ce type de réseau. Il est donc
10 nécessaire de prévoir une pièce de logiciel additionnelle, implantée dans le terminal 1, généralement sous la forme de ce qui est appelé un "plug-in", selon la terminologie anglo-saxonne. Cette pièce de logiciel, qui porte la référence 12 sur la figure 1A, effectue l'interface entre le navigateur 10 et la carte 2, plus précisément les circuits électroniques 20 de cette carte 2.

15 Dans l'état actuel de la technique, le système hôte associé au lecteur de carte 3, c'est-à-dire le terminal 1, est associé également à une application particulière. En d'autres termes, il est nécessaire de prévoir un terminal spécifique, dit "dédié", pour chaque application particulière.

En outre, il est clair que, même compte tenu de la rapide évolution passée des
20 technologies et de leur évolution future prévisible, la capacité d'enregistrement d'informations dans des circuits de mémoire, vive ou fixe, d'une carte à puce reste et restera très limitée, si on compare cette capacité à celle offerte par un terminal "hôte" de cette carte à puce, et naturellement à celles offertes par des systèmes plus importants, "mini-ordinateurs" ou grands systèmes de type dit
25 "main frame". Aussi, il n'est pas possible de stocker les données d'un nombre important d'applications dans une carte à puce, et notamment des fichiers de type multimédia très volumineux.

L'invention vise à pallier les inconvénients des dispositifs de l'art connu et dont certains viennent d'être rappelés, tout en répondant aux besoins qui se font
30 sentir. Il doit notamment être possible d'accéder à un grand nombre d'applications, même volumineuses d'un point de vue quantité de données, de

natures diverses et réparties sur tout le réseau Internet. En outre, dans un mode de réalisation préféré, les accès doivent bénéficier d'une sécurité maximale, c'est-à-dire dans la pratique s'effectuer via et sous le contrôle d'une carte à puce contenant toutes les données nécessaires à la sécurisation des échanges de données. Enfin, ces accès doivent pouvoir s'effectuer à partir d'un navigateur du commerce et être transparents pour un utilisateur, qui ne doit "voir" que la carte à puce comme unique interlocuteur, quel que soit le lieu de stockage de l'application.

Selon une première caractéristique du procédé, la carte à puce présente au système hôte, c'est-à-dire le terminal, un modèle de terminal virtuel, par exemple sous la forme d'une page en langage "HTML" (pour "HyperText Markup Language"), ou plus généralement en langage hypertexte, ou encore sous la forme d'un "applet", en langage "JAVA" (marque déposée), ce qui permet à l'utilisateur de choisir une application particulière parmi celles disponibles et proposées par la carte à puce. De ce fait, le terminal se trouve donc banalisé et supporte une pluralité d'applications. Le système hôte est vu comme un périphérique de la carte à puce, et il met à sa disposition des ressources matérielles, tels un écran de visualisation, un clavier, etc.

Pour ce faire, on prévoit une couche de logiciel de communication spécifique dans la carte à puce et son pendant dans le terminal. Le terme "spécifique" doit être entendu comme spécifique au procédé de l'invention. En effet, ces couches de communications, dites spécifiques, sont banalisées quelle que soit l'application considérée. Elles n'interviennent que dans le processus d'échange de données bidirectionnel entre la carte à puce et le terminal, d'une part, et la carte à puce et le réseau.

Les couches logicielles de communication spécifiques comprennent notamment des composants logiciels, dits "agents intelligents", permettant en particulier des conversions de protocoles. Il existe des agents intelligents appareillés dans les couches de communication spécifiques respectives associées au terminal et à la carte à puce. Selon le procédé de l'invention, il s'établit des sessions entre agents intelligents appareillés.

Selon une deuxième caractéristique, le procédé de l'invention rend possible l'activation d'applications de type conventionnel, c'est-à-dire du type "CGA" précité, localisées dans une carte à puce, sans devoir les modifier en quoi que ce soit.

5 Pour ce faire, on prévoit un ou plusieurs agents intelligents dits traducteurs de script, qui reçoivent des requêtes d'un navigateur et les traduisent en ordres "APDU" compréhensibles par l'application de type "CGA". Cette caractéristique technique permet d'implanter dans une carte à puce, dont l'architecture est conforme au procédé de l'invention, un mécanisme similaire à
10 la fonction dite "CGI" (pour "Common Gateway Interface") implantée dans les serveurs "WEB" classique.

Enfin, selon une autre caractéristique du procédé de l'invention, en mettant en œuvre les fonctions et mécanismes précités, celui-ci permet d'accéder à des ressources informatiques réparties sur un réseau de
15 transmission de données auquel est connecté le terminal, notamment le réseau Internet ou un réseau de type équivalent (intranet, extranet), sans que l'utilisateur ait à se soucier de leurs emplacements. Dans ce qui suit, comme il a été indiqué, ces ressources seront appelées "objets virtuels", statiques ou dynamiques.

20 Pour ce faire, il est mis en œuvre un agent intelligent traducteur de script dédié à cette tâche, coopérant avec les autres agents intelligents présents dans le terminal et/ou la carte à puce. Cet agent permet de définir les objets virtuels auxquels la carte à puce peut accéder, et de ce fait également l'utilisateur (ou porteur de la carte à puce), d'une part, et fournit au navigateur
25 interrogateur, via la carte à puce, des méthodes permettant d'accéder à ces objets virtuels d'autre part.

L'invention concerne donc un système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers
30 d'un terminal, caractérisé en ce que :

-il stocke au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance de cet objet ;

- 5 -il comprend des moyens d'interface de réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué constitue un serveur d'information sur le réseau ; et

- 10 -il comprend des moyens d'interface de fichier d'objet, agencés pour établir une correspondance entre des informations transitant par les moyens d'interface de réseau et affectées à au moins ledit fichier d'objet, et des informations échangées avec ledit fichier d'objet.

- 15 Avantageusement, le fichier d'objet comprend une pièce de logiciel autonome exécutable sur un logiciel de navigation. Avantageusement, cette pièce de logiciel autonome est capable de mettre en œuvre un système de gestion de fichier d'objet du système embarqué.

Avantageusement, ledit fichier d'objet comprend une description d'actions à réaliser pour instancier un objet. Avantageusement, les actions comprennent des actions réalisées à l'intérieur du système embarqué et consistant en des sessions entre agents du système embarqué.

- 20 Avantageusement, les actions comprennent des actions réalisées à l'extérieur du système embarqué et consistant en des sessions avec des agents du terminal en vue d'obtenir des informations à partir de serveurs d'information du réseau.

- 25 Avantageusement, lesdits moyens d'interface de réseau sont agencés pour coopérer avec les moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué se comporte comme un client capable de se connecter à au moins un serveur du réseau.

10

L'invention concerne aussi un procédé pour instancier un objet situé sur un réseau, et utilisant le système embarqué précité, caractérisé en ce qu'il permet de décrire un ensemble de sessions entre agents par un fichier d'objet, au moyen d'au moins les étapes suivantes :

- 5 -établir une liste des agents mis en œuvre ;

-pour chaque agent, définir des arguments d'appel, nécessaires à l'agent.

Avantageusement, un argument d'appel décrit l'ouverture d'une session avec un autre agent.

- Avantageusement, un agent modifie la liste des arguments utilisés par un autre agent.
10

En variante, le procédé est caractérisé en ce qu'il met en œuvre des sessions entre agents décrites par un fichier d'objet exécuté depuis le serveur d'information du système embarqué au moyen d'au moins les étapes suivantes :

- 15 -identification d'un fichier d'objet ;

-exécution de ce fichier d'objet.

Avantageusement, l'identification est effectuée par un nom de répertoire particulier.

- Avantageusement, l'identification est effectuée par un attribut de fichier
20 particulier.

Avantageusement, l'identification est effectuée par une convention de nommage particulière.

Avantageusement, l'exécution est effectuée par instanciation du premier agent associé au fichier objet.

Avantageusement, l'exécution est effectuée par instanciation d'un ou plusieurs agents référencés par le fichier objet.

En variante, le procédé est caractérisé en ce qu'il met en œuvre des sessions entre agents décrites par un fichier d'objet exécuté depuis un logiciel de navigation au moyen d'au moins les étapes suivantes :

-chargement par le logiciel de navigation d'un fichier d'objet et d'un logiciel spécifique capable de le mettre en œuvre ;

-exécution du logiciel spécifique par le logiciel de navigation.

Avantageusement, la réalisation du logiciel spécifique est effectuée au moyen de tout langage interprété, exécutable par le logiciel de navigation.

Avantageusement, l'interpréteur de fichier objet est réalisé sur un logiciel de navigation.

En variante, le procédé est caractérisé en ce qu'il permet au système embarqué de rendre possible la mise en œuvre de sessions entre agents décrites par un fichier d'objet exécuté depuis un logiciel de navigation, et en ce qu'il comprend l'étape consistant à identifier, au moyen d'un identificateur universel de ressource, un logiciel spécifique agencé pour mettre en œuvre le logiciel de navigation.

Avantageusement, l'identificateur universel de ressource est intégré à un document hyper-texte.

Avantageusement, ledit document hyper-texte est contenu dans le système embarqué.

Avantageusement, ledit document hyper-texte est contenu sur un serveur d'information du réseau, distant du système embarqué.

Avantageusement, ledit logiciel spécifique est chargé par une méthode disponible sur le logiciel de navigation et déduite de l'identificateur universel de ressource.

L'invention concerne aussi un système embarqué, équipé d'une puce
5 comprenant des moyens de traitement d'information et des moyens de
mémorisation d'information, et destiné à coopérer avec un réseau au travers
d'un terminal, caractérisé en ce qu'il comprend des moyens d'interface de
réseau, agencés pour coopérer avec des moyens d'interface de réseau
appariés, situés dans le terminal, de façon que le système embarqué constitue
10 un serveur d'information sur le réseau et/ou se comporte comme un client
capable de se connecter à au moins un serveur du réseau. L'invention
concerne aussi un terminal destiné à coopérer avec un réseau et comprenant
des moyens de traitement d'information, des moyens de mémorisation
d'information, et des moyens de coopération avec un système embarqué,
15 équipé d'une puce comprenant des moyens de traitement d'information et des
moyens de mémorisation d'information, caractérisé en ce qu'il comprend des
moyens d'interface de réseau, agencés pour coopérer avec des moyens
d'interface de réseau appariés, situés dans le système embarqué de façon que
le système embarqué constitue un serveur d'information sur le réseau et/ou se
20 comporte comme un client capable de se connecter à au moins un serveur du
réseau. Avantageusement, le terminal acquiert dynamiquement lesdits moyens
d'interface de réseau par un mécanisme de chargement de logiciel depuis le
réseau. Il peut s'agir notamment d'un mécanisme de « plug-in », selon la
terminologie anglo-saxonne. Si le terminal ne connaît pas l'extension du fichier
25 contenant ledit logiciel, il va chercher sur un serveur le logiciel associé à cette
extension, appelé usuellement « helper » selon la terminologie anglo-saxonne.
Avantageusement, lesdits moyens d'interface de réseau appariés constituent,
dans le terminal et dans le système embarqué, une pile comprenant une ou
plusieurs couches de communication telles qu'elles permettent au système
30 embarqué de partager tout ou partie des couches de communication du
terminal. Par ailleurs, le terminal dispose avantageusement de points d'accès

sur ses couches de communication, qui lui permettent de dériver un flux d'information depuis ou vers une ou plusieurs de ces couches. Ces points d'accès correspondent aux points connus en soi sous le nom « SAP » défini par la norme ISO (Service Acces Point, selon la terminologie anglo-saxonne).

5

L'invention concerne aussi un système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, caractérisé en ce qu'il comprend des moyens d'interface de
10 réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon qu'au moins une partie d'un flux d'information échangé entre une application du terminal et le réseau traverse les moyens d'interface de réseau du système embarqué, en fonction de critères connus du terminal. L'invention concerne aussi un terminal destiné à coopérer
15 avec un réseau et comprenant des moyens de traitement d'information, des moyens de mémorisation d'information, et des moyens de coopération avec un système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, caractérisé en ce qu'il comprend des moyens d'interface de réseau, agencés pour coopérer avec
20 des moyens d'interface de réseau appariés, situés dans le système embarqué de façon qu'au moins une partie d'un flux d'information échangé entre une application du terminal et le réseau traverse les moyens d'interface de réseau du système embarqué, en fonction de critères connus du terminal. La dérivation d'une partie du flux d'information vers le système embarqué est
25 avantageusement effectuée par les moyens de traitement d'information du terminal en fonction de critères pré-établis soit de manière statique, soit de manière négociée par un dialogue avec le système embarqué ; dans ce dernier cas, le terminal peut par exemple demander au système embarqué son adresse « IP » (s'il s'agit du réseau Internet), par des protocoles connus.
30 Avantageusement, les critères précités comprennent l'un des suivants :
-l'adresse « IP » du système embarqué, ou son adresse « ATM » dans le cas

d'un réseau ATM ;

-l'adresse IP du terminal et un port « TCP » ou « UDP » particulier ;

-tout point d'accès SAP qui référence le système embarqué.

- 5 L'invention va maintenant être décrite de façon plus détaillée en se référant aux dessins annexés, parmi lesquels :
- les figures 1A et 1B illustrent schématiquement les architectures matérielles et logiques, respectivement, d'un exemple de système d'application à base de carte à puce selon l'art connu ;
- 10 - la figure 2 illustre schématiquement un exemple de système d'application à base de carte à puce selon l'invention, cette dernière agissant en tant que serveur "WEB" ;
- la figure 3 illustre de façon simplifiée l'architecture logique d'un système dans lequel la carte à puce comprend des agents intelligents ;
- 15 - la figure 4 illustre une architecture de système conforme à l'invention, dans laquelle la carte à puce comprend des agents intelligents traducteurs de scripts ;
- la figure 5 est un diagramme illustrant schématiquement les principales phase d'échanges entre un navigateur et une carte à puce présentant
- 20 l'architecture de la figure 4 ;
- la figure 6 est un diagramme illustrant schématiquement un aspect essentiel du procédé selon l'invention par lequel il est possible d'accéder à des objets virtuels répartis sur un réseau de type Internet via une carte à puce et un navigateur de type "WEB" ;
- 25 - la figure 7 illustre schématiquement l'organisation d'un système de gestion de fichiers dit virtuel pour la mise en œuvre de cet aspect du procédé de l'invention ;
- la figure 8 est un exemple d'architecture comprenant un système de gestion de fichier virtuel selon la figure 7 ;
- 30 - les figures 9 à 15 sont des diagrammes illustrant schématiquement plusieurs mode de réalisation du procédé selon l'invention.

Avant de décrire le procédé d'activation d'applications localisées dans une carte à puce selon l'invention et de détailler une architecture pour sa mise en œuvre, il apparaît tout d'abord utile de rappeler brièvement les caractéristiques principales des protocoles de communication sur les réseaux.

- 5 L'architecture des réseaux de communication est décrite par diverses couches. A titre d'exemple, le standard "OSI" ("Open System Interconnection"), défini par l' "ISO", comporte sept couches qui vont des couches dites basses (par exemple la couche dite "physique" qui concerne le support de transmission physique) aux couches dites hautes (par exemple la couche dite d'
- 10 "application"), en passant par des couches intermédiaires, notamment la couche dite de "transport". Une couche donnée offre ses services à la couche qui lui est immédiatement supérieure et requiert de la couche qui lui immédiatement inférieure d'autres services, via des interfaces appropriées. Les couches communiquent à l'aide de primitives. Elles peuvent également
- 15 communiquer avec des couches de même niveau. Dans certaines architectures, l'une ou l'autre de ces couches peut être inexistante.

Dans un environnement de type Internet, les couches sont au nombre de cinq, et de façon plus précise, en allant de la couche supérieure à la couche inférieure : la couche d'application ("http", "ftp", "e-mail", etc.), la couche de

20 transport ("TCP"), la couche d'adressage de réseau ("IP"), la couche de liens de données ("PPP", "Slip", etc.) et la couche physique.

Ces rappels étant effectués, on va maintenant décrire une architecture de système d'application à base de carte à puce autorisant celle-ci à agir comme un serveur "WEB". Un exemple d'une telle architecture est représenté

25 schématiquement sur la figure 2. Les éléments communs aux figures 1A et 1B portent les mêmes références et ne seront re-décrits qu'en tant que de besoin. Pour simplifier le dessin, on n'a pas représenté les divers périphériques connectés au terminal (figure 1A : écran 5 et clavier 6, par exemple).

A l'exception de couches logicielles de protocole de communication

30 spécifiques, référencées 13 et 23a, respectivement implantées dans le terminal

1 et la carte à puce 2a, les autres éléments, matériels ou logiciels, sont communs à l'art connu.

Le terminal 1 comprend des circuits 11 d'accès au réseau *R/I*, constitués par exemple d'un modem pour le réseau Internet ou d'une carte réseau pour un
5 réseau local. Ces circuits regroupent les couches logicielles inférieures C1 et C2, correspondant aux couches "physique" et de "lien de données".

On a également représenté les couches supérieures C3 et C4, correspondant aux couches "d'adressage de réseau" ("IP", dans le cas d'Internet) et de "transport" ("TCP"). La couche supérieure d'application ("http", "ftp", "e-mail",
10 etc.) n'a pas été représentée.

L'interface entre les couches inférieures, C1 et C2, et les couches supérieures, C3 et C4, est constituée par une couche logicielle généralement appelée "driver couches basses". Les couches supérieures, C3 et C4, s'appuient sur cette interface et sont mises en œuvre par l'intermédiaire de bibliothèques de
15 fonctions spécifiques ou bibliothèques réseau 14, avec lesquelles elles correspondent. Dans le cas du réseau Internet, "TCP/IP" est mis en œuvre au moyen de bibliothèques dites de "sockets".

Cette organisation permet à un navigateur 10 (figure 1A) de poser des requêtes vers un serveur 4 (figure 1A), pour la consultation de pages "WEB"
20 (protocole "HTTP"), pour le transfert de fichiers (protocole "FTP") ou l'envoi de courrier électronique (protocole "e-mail"), ce de façon tout à fait classique.

Le terminal 1 comprend également un lecteur de carte 3, intégré ou non. Pour communiquer avec la carte à puce 2a, le lecteur de carte englobe également deux couches basses, CC1 (couche physique) et CC2 (couche de lien de
25 données), jouant un rôle similaire aux couches C1 et C2. Les interfaces logicielles avec les couches CC1 et CC2 sont décrites, par exemple, par la spécification "PC/SC" ("part 6, service provider"). Les couches elles-mêmes, CC1 et CC2, sont notamment décrites par les normes ISO 7816-1 à 7816-4, comme il a été rappelé.

Une couche logicielle supplémentaire 16 forme interface entre les couches applicatives (non représentées) et les couches inférieures, CC₁ et CC₂. La fonction principale dévolue à cette couche est une fonction de multiplexage/démultiplexage.

- 5 Les communications avec la carte à puce 2a s'effectuent selon un paradigme similaire à celui utilisé pour la manipulation de fichiers dans un système d'exploitation du type "UNIX" (marque déposée) : OUVRIR ("OPEN"), LIRE ("READ"), ECRIRE ("WRITE"), FERMER ("CLOSE"), etc.

Du côté de la carte à puce 2a, on retrouve une organisation similaire, à savoir
10 la présence de deux couches basses, référencées CCa₁ (couche physique) et CCa₂ (couche de lien de données), ainsi qu'une couche d'interface 26a, tout à fait similaire à la couche 16.

Selon une première caractéristique, on prévoit, de part et d'autre, c'est-à-dire dans le terminal 1 et dans la carte à puce 2a, deux couches de protocoles
15 spécifiques : 13 et 23a, respectivement.

Dans le terminal 1, la couche spécifique 13 s'interface aux "drivers couches basses" 15, aux bibliothèques 14 des couches réseau, C₃ et C₄, et aux couches protocolaires du lecteur de carte 3, c'est-à-dire les couches inférieures, CC₁ et CC₂, via la couche de multiplexage 16. La couche
20 spécifique 13 permet le transfert des paquets réseaux de et vers la carte à puce 2a. En outre, elle adapte les applications existantes telles le navigateur Internet 10 (figure 2), le courrier électronique, etc., pour des utilisations mettant en œuvre la carte à puce 2a.

Du côté de la carte à puce 2a, on retrouve une organisation tout à fait similaire
25 constituée par une instance supplémentaire de la couche spécifique, référencée 23a, pendant de la couche 13.

De façon plus précise, les couches spécifiques, 13 et 23a, sont subdivisées en trois éléments logiciels principaux :

- un module, 130 ou 230a, de transfert de blocs d'informations entre les
30 couches 13 et 23a, via les couches conventionnelles CC₁, CC₂, CCa₁ et CCa₂

- une ou plusieurs pièces de logiciel, dites "agents intelligents", 132 ou 232a, qui réalisent, par exemple, des fonctions de conversion de protocoles ;

- et un module de gestion de la configuration spécifique, 131 et 231a, respectivement ; module qui peut être assimilé à un agent intelligent particulier.

5 On retrouve donc, dans le terminal 1 et la carte à puce 2a, une pile protocolaire de communication entre les deux entités.

Les couches de niveau deux (couches de lien de données), CC2 et CCa2, assurent l'échange entre la carte à puce 2a et le terminal 1. Ces couches sont responsables de la détection et l'éventuelle correction d'erreurs de
10 transmission. Différents protocoles sont utilisables, et à titre d'exemples non exhaustifs les suivants :

- la recommandation ETSI GSM 11.11 ;
- le protocole défini par la norme ISO 7816-3, en mode caractère T=0 ;
- le protocole défini par la norme ISO 7816-3, en mode bloc T=1 ;

15 - ou le protocole défini par la norme ISO 3309, en mode trame "HDLC" (pour "High-Level Data Link Control procedure" ou procédure de commande de liaison à haut niveau).

Dans le cadre de l'invention, on utilisera de préférence le protocole ISO 7816-3, en mode bloc.

20 De façon connue en soi, à chaque couche de protocole, il est associé un certain nombre de primitives qui permettent les échanges de données entre couches de même niveau et d'une couche à l'autre. A titre d'exemple, les primitives associées à la couche de niveau deux sont du type "demande de données" ("*Data.request*") et "envoi de données" par la carte ("*Data.response*"),
25 ainsi que "confirmation de données" ("*Data.confirm*"), etc.

De façon plus spécifique, les couches 13 et 23a sont chargées du dialogue entre la carte à puce 2a et l'hôte, c'est-à-dire le terminal 1. Ces couches permettent l'échange d'informations entre un utilisateur (non représenté) du terminal 1 et la carte à puce 2a, par exemple via des menus déroulants sous la
30 forme d'hypertexte au format "HTML". Elles permettent aussi la mise en place

d'une configuration adaptée pour l'émission et/ou la réception de paquets de données.

Comme il a été indiqué ci-dessus, les couches comprennent trois entités distinctes.

- 5 La première couche, 130 ou 230a, est essentiellement constituée par un multiplexeur logiciel. Elle permet l'échange d'informations entre la carte à puce 2a et le terminal hôte 1, sous la forme d'unités de données de protocole. Elle joue un rôle similaire à celui d'un commutateur de paquets de données. Ces unités sont émises ou reçues via la couche de niveau 2 (couche de liens de
- 10 données). Ce protocole particulier de communication permet de mettre en communication au moins une paire d' "agents intelligents". Le premier agent de chaque paire, 132, est situé dans la couche 13, côté terminal 1, le second, 232a, est situé dans la couche 23a, côté carte à puce 2a. Une liaison entre deux "agents intelligents" est associée à une session. Une session est un
- 15 échange de données bidirectionnel entre ces deux agents.

Un agent intelligent peut réaliser tout ou partie de fonctions des couches de niveau trois et quatre, en fonction de la configuration mise en œuvre par le terminal 1.

- Un agent intelligent particulier est identifié avantageusement par un nombre
- 20 entier, par exemple sur 16 bits (nombre compris entre 0 et 65535). Cet identificateur est utilisé, par exemple, dans une unité de donnée de protocole constituant une référence de destination et une référence de source.

- Il existe deux grandes catégories d'agents intelligents : les agents de type "serveur", qui sont identifiés par une référence fixe, et les agents de type
- 25 "client", qui sont identifiés par une référence variable, délivrée par le module de gestion de configuration, 131 ou 231a.

- Le processus d'ouverture d'une session est habituellement le suivant : un agent intelligent de type "client" ouvre la session vers un agent intelligent de type "serveur". Les couches 130 et 230a gèrent des tables (non représentées)
- 30 qui contiennent la liste des agents intelligents présents, côté terminal hôte 1 et carte à puce 2a.

Les agents intelligents sont associés à des propriétés ou attributs particuliers. Pour fixer les idées, et à titre d'exemple non limitatif, les six propriétés suivantes sont associées aux agents intelligents :

- "hôte" : agent localisé dans le terminal ;
- 5 - "carte" : agent localisé dans la carte à puce ;
- "local" : agent ne communiquant pas avec le réseau ;
- "réseau" : agent communiquant avec le réseau ;
- "client" : agent qui initialise une session ;
- "serveur" : agent qui reçoit une demande de session.

- 10 Les agents intelligents permettent d'échanger des données (de l'hypertexte par exemple), mais également de déclencher des transactions réseau.

Les modules de gestion de configuration, 131 et 231a, respectivement, sont assimilables, comme il a été indiqué, à des agents intelligents particuliers. Par exemple, le module 131, côté terminal hôte 1, gère notamment des
15 informations relatives à la configuration de ce terminal (modes de fonctionnement), liste des autres agents intelligents présents, etc. Le module 231a, côté carte à puce 2a, a des fonctions analogues. Ces deux agents intelligents peuvent être mis en communication l'un avec l'autre pour établir une session.

- 20 Selon une caractéristique, la carte à puce 2a propose au système hôte, c'est-à-dire au terminal 1, un modèle de terminal virtuel. Pour ce faire, la carte à puce 2a se comporte comme un serveur "WEB".

La carte à puce 2a est "adressée" par le navigateur 10. Elle lui transmet alors une page de type "WEB" en langage "HTML", un "applet" ou toute autre pièce
25 de logiciel. A titre d'exemple, la page "WEB" peut se présenter sous la forme d'une page d'accueil donnant un choix d'applications possibles et/ou d'hyperliens vers des serveurs extérieurs.

De façon pratique, la carte à puce 2a est avantageusement "adressée" par utilisation d'une adresse "URL" (pour "Universal Resource Locator") définissant
30 un rebouclage sur le terminal 1 lui-même, et non un pointage sur un serveur

externe. A titre d'exemple, la structure de cette "URL" est habituellement la suivante :

http://127.0.0.1:8080 (1),

5 dans laquelle 127.0.0.1 est l'adresse "IP" de rebouclage et 8080 est le numéro de port.

La figure 3 illustre de façon simplifiée l'architecture logique d'un système dont la carte à puce 2a comprend des agents intelligents, dont deux seulement ont été représentés : un agent intelligent de type non précisément défini 232a2 et un agent intelligent 232a1, de type dit "WEB". La pile logique comprend, les
10 couches de protocole inférieures, référencées 200a, répondant aux normes ISO 7816-3 (figure 2 : CCa1 et CCa2), le gestionnaire de commandes "APDU" 201a1, et le multiplexeur de paquets 230a, ce dernier étant interfacé aux agents intelligents, notamment l'agent intelligent "WEB" 232a1.

Du côté terminal, il existe deux piles, l'une communiquant avec le réseau
15 Internet RI, l'autre avec la carte à puce 2a. La première pile comprend les organes 11 (figure 2 : C1 et C2) d'accès au réseau (normes OSI 1 et 2) et les couches de protocole "TCP/IP" (figure 2 : C3 et C4), référencées 100. Ces dernières couches sont interfacées avec le navigateur "WEB" 10. L'autre pile comprend les couches de protocole inférieures, référencées 101, répondant
20 aux normes ISO 7816-3 (figure 2 : C1 et C2), le gestionnaire 102 d'ordres "APDU" et le multiplexeur de paquets 130, ce dernier étant interfacé avec des agents intelligents, dont un seul 132, est représenté. Ce dernier, que l'on supposera de "type réseau", peut en outre communiquer, d'une part avec le navigateur 10, via les couches "TCP/IP" 100, d'autre part avec le réseau
25 Internet RI, via ces mêmes couches "TCP/IP" 100 et l'organe 11, d'accès au réseau RI.

Le gestionnaire d'ordres "APDU" 201a est également interfacé avec une ou plusieurs couches de niveau applications, que l'on appellera simplement applications. Ces applications sont, comme il a été indiqué, des applications de
30 type conventionnel, que l'on a appelé "cardlet".

En résumé, la fonction "serveur WEB", fournie par la carte à puce 2a, peut être réalisée par l'association de l'agent intelligent "WEB" 232a1 dans la carte à puce et de l'agent réseau 132 dans le terminal 1.

La carte à puce 2a présente donc bien la fonctionnalité serveur "WEB". En outre, selon une caractéristique du procédé de l'invention, n'importe quelle application conventionnelle, A_1 à A_n , du type "CGA" précité, peut être activée au travers de ce serveur "WEB", soit par le navigateur "WEB" 10 présent dans le terminal 1, soit par un navigateur éloigné, localisé en un point quelconque du réseau Internet R_I . Selon le procédé de l'invention, les applications, A_1 à A_n , ne nécessitent pas d'être ré-écrites et sont mises en œuvre telles quelles.

Selon une autre caractéristique de l'invention, ces applications restent accessibles à un terminal de type conventionnel, c'est-à-dire conforme à l'art connu.

Pour répondre à ces exigences, la fonction serveur "WEB", offerte par la carte à puce 2a, inclut un mécanisme similaire à la fonction dite "CGI" (pour "Common Gateway Interface") implantée dans les serveurs "WEB" classiques.

Avant de décrire un exemple d'architecture conforme à l'invention, permettant de réaliser une fonction de ce type, au sein même de la carte à puce, il est utile de rappeler les principales caractéristiques d'un mode de fonctionnement "CGI".

Le "CGI" est une spécification de mise en œuvre, depuis un serveur "WEB", d'applications écrites pour les systèmes d'exploitation "UNIX" (marque déposée), "DOS", ou "WINDOWS" (marque déposée). A titre d'exemple, pour le système d'exploitation "UNIX", la spécification est "CGI 1.1" et pour le système d'exploitation "WINDOWS 95", la spécification est "CGI 1.3".

Toujours à titre d'exemple, une requête "HTTP" du type :

"http://www.host.com/cgi-bin/xxx.cgi" (2),

dans laquelle "host" se réfère à un système hôte (généralement éloigné), est interprétée par un serveur "WEB" comme l'exécution d'un script de commande, de type "CGI" nommé "xxx" et présent dans le répertoire "cgi-bin" de ce système hôte. Bien que le nom du répertoire puisse être *a priori* quelconque,

par convention, c'est le nom donné au répertoire stockant les scripts de type "CGI". Un script est une suite d'instructions du système d'exploitation du système hôte dont le résultat final est transmis au navigateur "WEB" émetteur de la requête précitée. Différents langages peuvent être utilisés pour écrire ce script, par exemple le langage "PERL" (marque déposée).

De façon pratique, la requête est habituellement affichée sur un écran informatique sous la forme d'un formulaire compris dans une page "HTML". Le langage "HTML" permet de traduire un formulaire en une adresse "URL". Le formulaire comporte un ou plusieurs champs, obligatoires ou non, qui sont remplis par un utilisateur à l'aide des moyens de saisie habituels : clavier pour le texte, souris pour les cases à cocher ou les boutons dits "radio", etc. Le contenu du formulaire (ainsi qu'éventuellement des informations et instructions dites "cachées") est émis à destination du serveur "WEB". Le code "HTML" de la page décrit la structure matérielle du formulaire (cadre, graphisme, couleur, et tout autre attribut), ainsi que la structure des champs de données à saisir (nom, longueur, type de données, etc.).

La transmission peut s'effectuer selon deux types de formats principaux. Un premier format utilise la méthode dite "POST" et un second la méthode dite "GET". Une information de type de format est présente dans le code de la page formulaire.

Ce mécanisme n'est cependant pas directement transposable à une carte à puce, même si celle-ci offre la fonctionnalité serveur "WEB" conformément à l'une des caractéristiques de l'invention.

On va maintenant décrire un exemple d'architecture permettant d'activer une application quelconque, de type conventionnel, via un serveur "WEB" sur la carte à puce 2a, par référence à la figure 4.

Lors d'une première étape, un utilisateur (non représenté) invoque depuis son navigateur "WEB" (figure 3 : 10) une "URL" qui peut se présenter de la façon suivante :

dans laquelle "@carte" est une adresse IP de la carte à puce (par exemple l'adresse de rebouclage "127.0.0.1" précédemment décrite : voir formule (1)), et "xxx.html" est une page en langage "HTML" relative à une application particulière "xxx" offerte par la carte à puce.

- 5 Lors d'une deuxième étape, de la manière décrite précédemment, la carte à puce renvoie une page "HTML", par exemple de type formulaire.

Lors d'une troisième étape l'utilisateur renseigne les champs du formulaire et en transmet le contenu à la carte à puce, habituellement en cliquant sur un champ particulier, de type "bouton poussoir".

- 10 Les données sont alors émises et reçues par l'agent réseau 132. Les données traversent alors le multiplexeur de paquets 130 (qui constitue l'un des composants de la couche spécifique 13, côté terminal 1), le gestionnaire d'ordres "APDU" 102, les couches protocolaires 101, pour être transmises à la carte à puce 2a. Elles traversent ensuite les couches protocolaires 200a, le
15 gestionnaire d'ordres "APDU" 201a, le multiplexeur de paquets 230a pour être reçues par l'agent "WEB" 232a₁. Il s'établit donc une session logique entre les deux agents intelligents, comme explicité précédemment.

- Il convient de remarquer que les données adressées à l'agent "WEB" 232a₁ sont transportées, de façon conventionnelle en soi, sous formes d'ordres
20 "APDU" destinés à l'application particulière "Multiplexeur de paquets". Le gestionnaire d'ordres "APDU" 201a sélectionne cette application de manière tout à fait similaire aux autres applications de type "CGA" présentes dans la carte à puce 2a, référencées A₁ à A_n. En d'autres termes, le multiplexeur de paquets 230a est vu par le gestionnaire d'ordres "APDU" 201a comme une
25 application "CGA" ordinaire.

- La requête "HTTP" est analysée par l'agent "WEB" 232a₁ qui détecte une référence à un répertoire particulier, que l'on appellera ci-après par convention "cgi-smart", d'une part, et à une application particulière, par exemple "xxx" dans le cas de l'exemple décrit, d'autre part. Le chemin complet est donc, en
30 l'occurrence "cgi-smart/xxx".

Selon une caractéristique du procédé de l'invention, l'entité ci-dessus désigne un script particulier associé à une application également particulière "xxx".

Lors d'une quatrième étape, le script est interprété par un agent intelligent dit "Agent traducteur de script", que l'on appellera "ATS" ci-après. Cette traduction
5 peut être réalisée de différentes manières :

- a/ par l'agent "WEB" 232a₁ lui-même, qui est doté dans ce cas d'une double capacité ;
- b/ par un agent traducteur de script unique capable de traduire l'ensemble des scripts présents dans la carte à puce 2a ;
- 10 c/ par un agent de script dédié que l'on appellera "ATSD" ci-après (un par script) ; ou
- d/ par un agent "APDU" 2010a du gestionnaire d'ordres "APDU" 201a, qui est doté, dans ce cas, d'une double capacité.

L'agent "APDU" 2010a est une composante de la couche gestionnaire d'ordres
15 "APDU" 201a. Cette dernière, comme il a été indiqué, est une couche capable de centraliser tous les ordres "APDU" émis et/ou reçus par le système, de sélectionner des applications, parmi A₁ à A_n, mais également d'offrir une interface de type agent intelligent. Elle est donc capable, selon l'une des caractéristiques du procédé, de communiquer avec tous les agents intelligents
20 du système (via des sessions), que ces agents soient localisés dans le terminal 1 ou la carte à puce 2a.

Dans le cas c/ ci-dessus, une session est ouverte entre l'agent "WEB" 232a₁ et l'un des agents "ATSD".

La figure 4 illustre un exemple d'architecture pour laquelle les agents
25 traducteurs sont du type "ATSD". Ils sont référencés ATS₁ à ATS_n et associés aux applications A₁ à A_n. L'application sélectionnée étant supposée être l'application A_i, la session s'établit entre l'agent "WEB" 232a₁ et l'agent ATS_i.

Un agent traducteur de script génère une suite d'ordres "APDU". Une session est ouverte entre l'agent traducteur, par exemple l'agent ATS_i, et l'agent
30 "APDU" 2010a. Les ordres sont alors émis vers l'agent "APDU" 2010a. Le

gestionnaire d'ordres "APDU" 201a sélectionne l'application "CGA" A_j (par exemple l'application "PME") et lui transmet les ordres "APDU", ordres traduits et donc conventionnels, qu'elle est en mesure de comprendre. Cette application est donc correctement activée, sans avoir à la modifier ou à la réécrire.

- 5 Les réponses de l'application "CGA" A_j sont transmises au gestionnaire d'ordres "APDU" 201a, à l'agent "APDU" 2010a, puis de nouveau à l'agent ATS_j (et de façon plus générale à l'agent traducteur de script).

En fonction de la réussite ou de l'échec du déroulement du script, l'agent traducteur de script, par exemple l'agent ATS_j dans l'exemple de la figure 4,
10 élabore une page en langage "HTML" et la transmet via les différentes couches empruntées par la requête initiale, mais en sens inverse, ce pour être présentée sur l'écran de visualisation 5 (figure 1A).

Les différents cheminements sont représentés symboliquement sur la figure 4 par des traits pleins reliant les blocs fonctionnels ou en pointillés à l'intérieur de
15 ces blocs.

La figure 5 résume de façon schématique les principales étapes du processus qui vient d'être décrit :

- a/ transmission via le réseau Internet RI (ou à partir du terminal local : dans les deux cas à l'aide d'un navigateur conventionnel 10) d'une requête "HTTP",
20 référencée RQ ;
- b/ réponse du serveur "WEB" de la carte à puce 2a, sous la forme d'un formulaire, référencé FO ;
- c/ transmission du formulaire rempli, sous la forme d'une nouvelle requête RQ ; et
- 25 d/ réponse sous la forme d'une page "HTML", référencée PR.

La réponse pourrait d'ailleurs consister en la transmission d'un fichier, ou d'une pièce de logiciel ou "Applet".

En mettant en œuvre les mécanismes et fonctions qui viennent d'être décrits, notamment la fonction serveur "WEB" et le recours à des agents intelligents
30 traducteurs de scripts, selon une caractéristique essentielle, le procédé selon l'invention va permettre de définir un environnement virtuel, avantageusement

sécurisé par la carte à puce. Dans un mode de réalisation préféré, cet environnement est compatible avec les applications de type dit multimédia.

Cette dernière caractéristique est particulièrement avantageuse car les navigateurs "WEB" récents, mais de type tout à fait classique en soi, permettent, par nature, de construire des environnements multimédias (images animées, sons, etc.). Ils sont en effet associés à des outils logiciels, intégrés ou non, permettant de manipuler des fichiers multimédia (visionneuse, etc.). En tout état de cause, les navigateurs permettent de télécharger des fichiers de données multimédias, habituellement volumineux et de les stocker sur un disque dur, par exemple dans le terminal, ou sur un organe de stockage de masse similaire. On a notamment proposé des technologies permettant l'affichage en temps réel ou quasi-réel de séquences vidéos ou la reproduction de son, à partir de sites "WEB" du réseau Internet.

Cependant, une carte à puce, comme il a été rappelé, ne présente qu'une faible capacité de mémoire. En outre, elle n'autorise qu'un très faible débit de données lors des échanges. Il n'est donc pas possible d'enregistrer un grand nombre de fichiers de données dans celle-ci. Il n'est pas non plus envisageable, de façon pratique, de stocker des fichiers multimédias, à l'exception de très courtes séquences ou de séquences sonores codées sous un format particulier, tel que le codage "MIDI".

En dehors de ces limitations d'ordre technologique, il est aussi souhaitable de pouvoir avoir accès à des applications éloignées, tout en bénéficiant d'une sécurisation de haut niveau, que seule la mise en œuvre d'une carte à puce est apte à offrir.

Le procédé selon l'invention permet ce mode de fonctionnement. L'environnement virtuel multimédia sécurisé par carte à puce, selon un mode de réalisation préféré, permet de :

- définir des objets virtuels auxquels la carte à puce peut accéder ;
- fournir des méthodes d'accès à ces objets.

La figure 6 illustre schématiquement cet aspect essentiel du procédé selon l'invention.

Un utilisateur U_i interroge la carte à puce 2a grâce au navigateur "WEB" 10 compris dans le terminal 1. Selon un mécanisme qui va être précisé ci-après, grâce notamment à la fonction "serveur WEB" précédemment décrite, la carte à puce 2a va renvoyer au navigateur une liste d'objets dits virtuels Obv_i , i étant
5 un indice arbitraire, auquel il a accès, c'est-à-dire de façon pratique pour lesquels la carte à puce 2a ou l'utilisateur U_i possède des droits d'accès. En effet, ces droits d'accès peuvent être liés strictement à la carte à puce 2a et immuables. Ils peuvent également être liés à un profil utilisateur, l'utilisateur U_i fournissant, par exemple, des données d'identification et un mot de passe. La
10 carte à puce 2a effectue une vérification par comparaison avec des données d'une base de données de sécurité enregistrées dans une mémoire fixe et, si le résultat de la comparaison est positif, fournit une liste d'objets virtuels Obv_i associée au couple : "données d'identification - mot de passe". De façon connue en soi, cette première phase peut mettre en œuvre un procédé de
15 chiffrement des données échangées entre le terminal et la carte à puce 2a ou mettre en œuvre un protocole de transmission sécurisé "HTTPS". La carte à puce 2a va également fournir une liste de méthodes d'accès aux objets virtuels Obv_i .

Les objets virtuels Obv_i , qui sont de type statique ou dynamique comme
20 indiqué précédemment, peuvent être localisés indifféremment dans la carte à puce 2a, ou dans le terminal 1, ou, de façon plus générale, dans un système quelconque connecté au réseau Internet RI . Selon une caractéristique de l'invention, cet emplacement est "transparent" pour le navigateur 10, et donc pour l'utilisateur U_i , comme il va l'être montré.

25 Le procédé selon l'invention fait appel notamment à ce qui va être appelé ci-après un système de gestion de fichier virtuel, ou "SGFV", et un agent intelligent traducteur de script spécialisé, que l'on appellera "ATSDA/SGVF", dédié à cette tâche. Cet agent intelligent fournit la liste des objets virtuels Obv_i auxquels la carte à puce 2a peut accéder. Une adresse "URL" particulière est
30 associée à chaque objet virtuel Obv_i . L'invocation de cette "URL" depuis le

navigateur "WEB" 10 permet d'instancier l'objet virtuel *Obvj*, au moyen d'une méthode d'appel déterminée, spécifique ou non à cet objet.

On va tout d'abord rappeler brièvement les principales caractéristiques d'un système de gestion de fichiers classique, appelé ci-après "SGF". Un tel système est utilisé pour stocker de l'information sur un support tel qu'un disque dur. L'information est mémorisée sous la forme d'un fichier. Un fichier, que ce soit des données pures ou des instructions de programme, est composé classiquement d'une suite de blocs de taille fixe. Un mécanisme bien connu permet d'obtenir la liste des blocs de mémoire qui constitue le fichier et leurs adresses dans la mémoire.

Un répertoire est un fichier particulier dont le contenu est une liste de descripteurs de fichier. Un tel descripteur comprend par exemple les éléments suivants :

- le nom du fichier ;
 - 15 - la longueur du fichier ;
 - la date de création ;
 - une référence permettant de retrouver la liste des blocs du fichier (numéro du premier bloc, tableau des numéros de blocs, etc.) ; et
 - et des attributs qui spécifient des propriétés particulières du fichier
- 20 (répertoire, lecture, écriture, exécution, etc.).

Le premier répertoire est habituellement appelé répertoire racine. Un répertoire qui n'est pas racine est dit sous-répertoire. Le répertoire qui contient le descripteur d'un fichier donné est son répertoire père. L'adresse d'un fichier dans le "SGF" est donc une succession de noms de répertoires, depuis le répertoire racine jusqu'au répertoire père du fichier, ce qui définit un chemin. A titre d'exemple, un tel chemin se présente comme suit :

"/racine/répertoire1/répertoire2/nom_du_fichier" (4),

les chiffres 1 et 2 étant arbitraires, "racine" étant le nom du répertoire racine et "nom_de_fichier" un nom quelconque de fichier.

30 Pour une carte à puce, la norme ISO 7816-4 définit le répertoire racine dit "MF" (pour "Master File" ou fichier maître), des sous-répertoires dits "DF" (pour

"Dedicated Files" ou fichiers dédiés) et des fichiers élémentaires dits "EF" (pour "Elementary Files").

Dans le cadre de l'invention, le système de gestion de fichiers "SGFV", que l'on appellera virtuel, permet de définir des objets virtuels Obv_i auxquels la carte à puce 2a peut avoir accès. Selon le procédé de l'invention, un objet virtuel Obv_i est associé à un fichier élémentaire virtuel. Le contenu d'un fichier élémentaire virtuel est constitué par l'ensemble des informations qui permettent d'accéder à l'objet virtuel associé Obv_i et d'en obtenir une instance dans le terminal 1.

De façon pratique, comme illustré schématiquement par la figure 7, le système "SGFV" peut constituer un sous-ensemble d'un système "SGF" classique, et, de façon plus précise, un "SGFV" est logé à l'intérieur d'un fichier élémentaire, tel que définit par la norme ISO 7816-4 précitée.

Un descripteur de fichier comportera généralement les éléments suivants :

- le nom du fichier ;
- 15 - la longueur du fichier ;
- la date de création ;
- une référence (avantageusement un nombre entier) qui permet de retrouver la liste des blocs du fichier (numéro du premier bloc, tableau de numéros de blocs, etc.) : un fichier virtuel est identifié par son nom ou cette
- 20 référence unique ; et
- des attributs du fichier qui spécifient les références particulières du fichier : répertoire ou fichier élémentaire, virtuel ou non virtuel, direct ou indirect.

On appelle "objet virtuel direct", un objet qui est instancié depuis la carte à puce 2a. Il s'agit typiquement d'un objet virtuel Obv_i statique qui peut être manipulé par le navigateur, par exemple affiché (image, etc.). On appelle "objet virtuel indirect", un objet virtuel Obv_i qui est instancié depuis le navigateur 10, typiquement à l'aide d'un "applet".

La figure 8 illustre schématiquement l'architecture d'un système à carte à puce permettant d'instancier un objet virtuel Obv_i localisé à un endroit quelconque du réseau Internet R/I , via le navigateur 10 et la carte à puce 2a.

Les éléments communs aux figures précédentes portent les mêmes références et ne seront re-décrits qu'en tant que de besoin.

L'architecture illustrée sur la figure 8 est très similaire à celle de la figure 4. La différence essentielle est constituée par le fait que l'on a prévu un "SGFV" 8, stocké dans la carte à puce 2a, et un agent intelligent traducteur de script spécifique "ATSDA/SGFV", référencé 7. Le mode de fonctionnement est similaire à celui illustré par la figure 4 lorsque l'on désire accéder à une application particulière A_j . Il est donc inutile de le re-décrire en détail. Dans le cas présent l'application particulière est remplacée par le système de gestion de fichier virtuel "SGFV" 8. On établit tout d'abord une session entre l'agent intelligent réseau 132 et l'agent intelligent "WEB" 232a₁. Selon le mécanisme précédemment explicité, il s'établit ensuite une session entre l'agent "WEB" 232a₁ et l'agent intelligent "ATSDA/SGFV" 7.

De façon pratique, l'agent intelligent "ATSDA/SGFV" 7 est accessible par des "URL", typiquement du type :

"http://www.host.com/cgi-smart/sgfv?" (5),

dans lesquelles "sgfv" est une application de type "CGI" associée à l'agent intelligent "ATSDA/SGFV" 7. La requête ci-dessus permet de parcourir l'arbre des répertoires et de "montrer" leur contenu au navigateur 10, au moyen d'une page "HTML". Les "feuilles" de l'arbre sont des fichiers élémentaires, virtuels ou non virtuels, associés à un hyperlien. La transmission dans le sens "carte à puce 2a - terminal 1" est réalisée de la manière qui a été explicitée en regard de la figure 4.

En d'autres termes, l'agent intelligent "ATSDA/SGFV" 7 associe à tout élément du "SGFV" 8, répertoire ou fichier élémentaire, une adresse "URL". L'adresse "URL" d'un répertoire désigne une page "HTML" qui contient la liste de ses éléments. L'adresse "URL" d'un fichier élémentaire permet de créer une instance de l'objet virtuel Obv_j associé à ce fichier virtuel.

Pour fixer les idées, si on utilise l'adresse "URL" (5) ci-dessus, on obtient une page "HTML" qui présente le contenu du répertoire racine au navigateur 10. Ce

répertoire racine est constitué par un ensemble de sous-répertoires et de fichiers comme illustré schématiquement par la figure 9. Sur cette figure, on a représenté un répertoire racine *rep#0*, au niveau supérieur, un fichier élémentaire réel *fe#7* et un sous-répertoire réel *srep#1*, au niveau
5 immédiatement inférieur, et un sous-répertoire virtuel *rep#2* et un fichier élémentaire virtuel *fe#5*, au niveau le plus bas, tous deux dépendants du sous-répertoire réel *srep#1*, les numéros de référence étant purement arbitraires.

Lors d'une première phase, l'agent intelligent "ATSDA/SGFV" 7 transmet au navigateur 10, en réponse à la requête reçue, une page "HTML" (non représentée) montrant, sous une forme ou une autre, la structure hiérarchique du "SGFV" 8. La page est habituellement affichée sur un écran de visualisation (figure 1A : 5), par exemple sous la forme d'un menu. Chaque ligne du menu est constituée d'un hyperlien décrivant un sous-répertoire ou un fichier élémentaire. L'affichage peut être avantageusement sous forme
15 graphique, associée ou non à un texte descriptif, le dessin de l'arbre de la figure 9 étant affiché sur l'écran précité. On peut encore afficher des icônes ou des formes complexes (par exemple en 3 dimensions), chacune étant associée à l'un des objets virtuels à instancier, et pouvant être représentative de leur nature (à titre d'exemple, une caméra représentant un fichier vidéo), associée
20 ou non à un texte descriptif.

L'utilisateur U_i est invité à cliquer sur un hyperlien (sur un nœud ou sur une branche dans le cas d'un graphique). Par cette action, il va pouvoir obtenir une instance de l'objet virtuel Obv_i désiré.

Le système "SGFV" 8 est avantageusement enregistré dans une mémoire de type re-programmable comprise dans la carte à puce 2a, par exemple du type
25 "EEPROM" (mémoire effaçable électriquement), comme illustré schématiquement sur la figure 10. Le "SGFV" 8 reproduit la structure de l'arbre de la figure 9.

Toujours dans l'exemple décrit, une fois obtenu la page de menu, obtenue lors d'une phase initiale, en cliquant sur une "URL" qui pourrait être
30 typiquement la suivante :

"http://www.host.com/cgi-smart/sgvf ?/file#5 (6),

l'utilisateur U_j obtient une instance de l'objet virtuel Obv_5 associé au fichier élémentaire référencé $fe\#5$ sur la figure 10. De même, il aurait pu obtenir le contenu d'un sous-répertoire, le paramètre "file#5" étant remplacé par "file#x" dans (6), #x étant le numéro associé au sous-répertoire.

Les fichiers non virtuels sont enregistrés dans la carte à puce 2a et sont conformes au paradigme usuel qui gouverne les "SGF". Ils contiennent des données, telles que par exemple des clés, données utiles à l'agent intelligent "ATSDA/SGFV" 7.

10 Différentes conventions sont possibles quant à la définition des informations nécessaires à l'instanciation d'un objet virtuel Obv_i , et par exemple :

- un fichier virtuel de longueur nulle hérite des méthodes d'accès de son répertoire père ;
- 15 - un répertoire virtuel est associé à un fichier élémentaire virtuel dont le nom est imposé (par exemple "virtual"), et qui contient les méthodes d'accès de ce répertoire.

En effet, outre la liste des objets virtuels accessibles Obv_i , un agent intelligent "ATSDA/SGFV" 7 doit également fournir une méthode d'accès à un objet virtuel donné Obv_i , ce à partir de tout ou partie des informations contenues dans un fichier virtuel élémentaire. La figure 11 illustre schématiquement ce processus.

Selon le procédé de l'invention, on prévoit deux méthodes d'accès, que l'on appellera directe et indirecte, respectivement, en fonction des attributs du fichier élémentaire virtuel considéré.

25 La méthode directe consiste en une description d'une chaîne d'agents intelligents mis en œuvre dans le processus permettant d'accéder à un objet virtuel Obv_i et d'en obtenir une instance dans le terminal. Lors de l'ouverture d'une session, un agent intelligent donné reçoit, de l'agent qui initie cette

session, une liste de structures d'appel qui sera dénommée ci-après "méthode d'appel" ou encore "Method PDU" (pour "Method Protocol Data Unit").

Une structure d'appel comprend :

- un identificateur de l'agent intelligent avec lequel la session est ouverte ;
- 5 - des données, ou argument, nécessaires à son utilisation.

Le premier agent intelligent adressé par la liste précitée "consomme" une première structure d'appel qui lui est destinée. Il transmet le reste de la liste de structure à un agent intelligent suivant, avec lequel il établit une session, jusqu'à épuisement de la liste.

- 10 Pour fixer les idées, un exemple des différentes étapes d'échanges entre l'agent intelligent "ATSDA/SGFV" 7 et deux agents intelligents en cascade, 232a_m et 232a_n, est illustré schématiquement par la figure 12. La liste de structure d'appel émise par l'agent intelligent "ATSDA/SGFV" 7 comporte en réalité deux sous-listes distinctes, repérées #1 et #2 dans leurs
- 15 en-têtes, respectivement. La première est consommée par le premier agent intelligent, 232a_m, et la seconde par le second agent intelligent, 232a_n. Un agent intelligent, par exemple l'agent intelligent 232a_m, est identifié par une référence, ou identificateur d'agent ("*Identificateur Agent #1*" ou "*Identificateur Agent #2*"). L'agent intelligent adressé, et avec lequel s'établit une session,
- 20 retient la sous-liste qui lui est destinée grâce à l'en-tête ("*Structure d'Appel #1*" ou "*Structure d'Appel #2*"). Les arguments de la sous-liste qu'il retient ("*Argument#1*" ou "*Argument#1*") sont constitués par un ensemble de données utiles au bon fonctionnement de cet agent. A titre d'exemple, une donnée peut être un nom de fichier (non virtuel ou virtuel direct).

- 25 Un agent intelligent donné, par exemple l'agent intelligent 232a_m, peut modifier le reste de la liste de structure d'appel avant de la transmettre à l'agent intelligent suivant, 232a_n. Pour ce faire, il adresse cet agent intelligent, 232a_n, et établit une session avec lui.

- 30 La méthode d'appel peut être avantageusement décrite au moyen du langage ASN.1 (« Abstract Syntax Notation 1 » de l'ISO)

La méthode d'accès directe permet en définitive d'instancier un objet virtuel *Obvj* directement depuis la carte à puce 2a. Il s'agit *a priori* d'un objet statique. L'objet instancié se présente habituellement sous la forme d'une page "HTML" ou d'un "applet" transmis au navigateur 10.

- 5 La seconde méthode d'accès, ou méthode d'accès indirect, est en réalité également une méthode d'accès direct, mais mise en œuvre à partir du terminal 1, et non plus de la carte à puce 2a. Cette méthode est essentiellement utilisée pour instancier des objets virtuels *Obvj* du type dynamique.

- 10 Selon cette variante du procédé, en réponse à un "URL" qui désigne un fichier élémentaire virtuel *fe#x*, l'agent intelligent "ATSDA/SGFV" 7 transmet au navigateur 10 une page "HTML" qui comporte un hyperlien pointant sur la méthode d'accès directe associée à l'objet virtuel *Obvj*

Deux variantes peuvent être mises en œuvre.

- 15 La première variante consiste à utiliser un "applet". Le lien sur la méthode d'accès est alors un "applet" localisé à l'adresse "@carte", qui peut elle-même être désignée par :

- le nom (c'est-à-dire une "URL") d'un fichier non virtuel, enregistré sur la carte à puce 2a ;
- 20 - une "URL" qui désigne un fichier virtuel direct.

- Un paramètre d'appel de cet "applet" est une liste de structure d'appel, par exemple codée en ASN.1 comme indiqué ci-dessus. L' "applet" contenu dans une page "HTLM" est téléchargé, depuis la carte à puce 2a ou le réseau Internet *RI*, vers le navigateur 10, puis exécuté par celui-ci de façon obligatoire (forçage). Cet "applet" établit une session avec un premier agent intelligent, arbitrairement référencé 232a_p. La connexion à cet agent intelligent 232a_p utilise, par exemple, un modèle d'échange de données du type client-serveur "TCP/IP" (c'est-à-dire la classe dite "socket JAVA"). L' "applet" se comporte comme un client "TCP/IP" et se connecte à un serveur "TCP/IP" (ce dernier étant également un agent intelligent) identifié par l'adresse de la carte et un
- 30 port : "@carte : port".

La figure 13 illustre schématiquement les différentes phases des échanges permettant d'instancier un objet virtuel par la méthode indirecte. On a repris sur cette figure 13 les paramètres de l'exemple précédemment décrit, en l'occurrence, le fichier élémentaire virtuel *fe#5*, ce qui se traduit par l'adresse "URL" de la configuration (6) ci-dessus. On a supposé que l'adresse de la carte à puce à utiliser est "@carte" et le port 8080. La requête est transmise à l'agent intelligent "ATSDA/SGFV" 7 selon le processus précédemment décrit. Celui-ci renvoie au navigateur 10 une page "HTML" *P* constituée par un "applet". Dans un but de simplification du dessin, les différentes instructions de cet "applet" ont été résumées sur la figure 13 par la mention "Code applet" placée entre les marqueurs <applet ...> et </applet>. De façon connue en soi, l' "applet" est associé à une classe "JAVA", que l'on a appelée arbitrairement "tv.class", pour "terminal virtuel". Le code comprend également des instructions indiquant l'adresse du premier agent intelligent de la structure de liste, référencé 232ap, et l'adresse et le port à utiliser, en l'occurrence l'adresse "@carte" et le port 8081. Il est à noter que cet agent intelligent 232ap peut être localisé dans la carte à puce 2a ou dans le terminal 1.

La phase suivante consiste, pour le navigateur, à demander l'applet à la carte 2a au moyen de la liste de structure d'appel, laquelle définit les paramètres d'appel de l'applet. En réponse, la carte lui transmet l'applet, lequel sera chargé par le navigateur sur sa machine virtuelle « Java », où il sera exécuté. La phase suivante consiste, pour le navigateur, à appeler l'agent intelligent 232ap en utilisant la classe « socket » du langage « Java ».

Chaque agent intelligent, par exemple 232ap, exécute une tâche précise : déchiffrement d'un message chiffré, vérification de mots de passe et/ou de données de sécurité, conversion d'un fichier depuis un premier format à un autre, etc. Bien que l'on n'ait représenté qu'un seul agent intelligent, 232ap, on peut prévoir, en tant que de besoin, plusieurs agents intelligents en cascade comme dans le cas précédent (figure 12), ce qui est illustré sur la figure 13 par les pointillés. Comme précédemment également chaque agent intelligent,

232a_p, consomme une partie de la structure de liste, celle qui lui est destinée, et transmet le reste, inchangé ou modifié, à l'agent intelligent suivant (non représenté).

Pour mieux illustrer la première variante, et pour fixer les idées, on va
5 considérer que l'utilisateur U_i désire télécharger et exécuter un fichier audio, codé par exemple au format "MP3". Ce fichier constitue l'un des objets virtuels, ici référencé FS , proposés par la page de menu "HTML" transmise par l'agent intelligent "ATSDA/SGFV" 7, lors de la phase initiale. La figure 14 illustre schématiquement la séquence d'étapes permettant d'instancier un tel objet
10 virtuel, référencé FS . On suppose que le navigateur 10 ne dispose pas d'un lecteur approprié pour un tel format. Ce lecteur, référencé LS , va être cherché sur un site Internet, qui peut être distinct ou non du site où se trouve le fichier son FS .

Dans l'exemple décrit, la séquence d'étapes est la suivante.

- 15 a/ l'utilisateur U_i clique sur un hyperlien (texte, icône ou toute autre représentation graphique de l'objet à rechercher, c'est-à-dire le fichier FS) : une requête I_1 est transmise à la carte à puce 2a ;
b/ en réponse, R_1 , une page "HTML" est transmise, par la carte à puce 2a, au terminal 1 et au navigateur 10 ;
- 20 c/ la page "HTML" reçue force le navigateur 10 à demander un "applet" : interrogation I_2 (dans le cas présent, il s'agit d'aller chercher le lecteur de son approprié LS) ;
d/ en réponse, R_2 , le lecteur recherché LS est téléchargé et installé dans le terminal 1 ;
- 25 e/ le navigateur 10 adresse de nouveau la carte à puce 2a, requête I_3 , en vue d'obtenir une instance du fichier audio FS ; et
f/ en réponse, le navigateur 10 reçoit ce fichier audio FS , ce dernier pouvant être lu, c'est-à-dire joué par le terminal 1, qui dispose désormais du lecteur de son LS approprié.

Il est à noter que toutes les opérations sont transparentes pour l'utilisateur U_i , plus précisément pour le navigateur 10, qui ne "connaît" que la carte à puce 2a. Le lecteur LS (ou de façon plus générale un autre "applet") et/ou l'objet virtuel recherché, c'est-à-dire le fichier FS dans l'exemple, si leurs tailles avaient été
5 compatibles avec la capacité de mémorisation de la carte à puce 2a, auraient d'ailleurs pu être enregistrés dans celle-ci (re-bouclages symbolisés par des traits en pointillés sur la figure 14). Le navigateur 10 ne connaît pas la localisation exacte des objets virtuels Obv_j . Seule la carte à puce 2a, de façon plus précise l'agent intelligent "ATSDA/SGFV" 7, connaît la localisation des
10 objets virtuels de la liste du "SGFV" 8 et la méthode pour y accéder.

Dans une variante préférée du procédé, l'agent intelligent "ATSDA/SGFV" 7 connaît également la liste des seuls objets virtuels accessibles à un utilisateur U_i donné (autorisations). Il s'agit donc bien d'un système sécurisé. Le terme "sécurisé" doit être considéré dans son sens le plus large. A titre d'exemple, il
15 concerne aussi bien des cartes à péage donnant accès à certaines ressources, en fonction d'un abonnement déterminé par exemple, ou des cartes assurant un accès sécurisé proprement dit à des ressources confidentielles, en fonction d'un niveau d'habilitation par exemple. Comme il a été indiqué, les ressources ou objets virtuels Obv_j peuvent être constituées par des transactions.

20 Ces remarques s'appliquent d'ailleurs aussi pour les méthodes d'accès direct. Cela constitue une caractéristique du procédé selon l'invention.

Selon une seconde variante, illustrée schématiquement par la figure 15, on peut utiliser un hyperlien qui définit l'adresse "TCP/IP" du premier agent intelligent associé à la méthode d'accès. L'adresse est du type :
25 "@Agent:AgentPort", avec "@Agent", l'adresse proprement dite de l'agent intelligent concerné et "AgentPort" le port de celui-ci. La liste "MethodPDU" est dans ce cas un paramètre d'une "URL". L'hyperlien sera associé, par exemple à une image ou un formulaire d'une page "HTML" P' .

Ainsi, à titre d'exemple, une "URL" ayant la structure suivante :

30 [http://@Agent:AgentPort/MethodPDU?Value=xx... \(7\),](http://@Agent:AgentPort/MethodPDU?Value=xx... (7),)

permet d'atteindre un agent intelligent qui se comporte comme un serveur "WEB TCP/IP", référencé arbitrairement 232a_q. Cet agent intelligent 232a_q est localisé à l'adresse "@Agent:AgentPort" et reçoit la liste de structure d'appel "MethodPDU", avec le paramètre "Value=xx...".

- 5 Pour fixer les idées, on suppose que l'objet virtuel *Obv_j* est une image à afficher sur un écran (figure 1A :5), d'un format particulier, à l'aide du navigateur 10 et que celui-ci ne dispose pas d'un programme approprié pour cet affichage, généralement appelé visionneuse ou "viewer" selon la terminologie anglo-saxonne. Il s'agit, par exemple, d'un programme exécutable
- 10 sous le système d'exploitation utilisé sur le terminal 1, de type "XXX.exe", avec "XXX" le nom de ce programme. L'action de cliquer sur l'hyperlien (7) ci-dessus va permettre de rechercher ce programme exécutable, celui-ci pouvant être localisé dans le terminal 1 ou dans un système éloigné.

- La différence entre les deux variantes de réalisation est que dans le premier
- 15 cas, le navigateur se voit "forcé" de demander le chargement d'un "applet". Toutes les étapes sont réalisées automatiquement. Dans le second cas, l'utilisateur *U_j* est invité à cliquer sur un hyperlien ou à exécuter une action similaire.

- A la lecture de ce qui précède, on constate aisément que l'invention atteint
- 20 bien les buts qu'elle s'est fixés.

Il doit être clair cependant que l'invention n'est pas limitée aux seuls exemples de réalisations explicitement décrits, notamment en relation avec les figures 2 à 15.

- En particulier, comme en ce qui concerne les autres agents intelligents
- 25 traducteurs de script; la fonction de l'agent intelligent associé au système de gestion de fichier virtuel peut être remplie par un agent non dédié : l'agent "WEB" ou l'agent "APDU".

REVENDICATIONS

- 1 Système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, caractérisé en ce que :
- 5 -il stocke au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance de cet objet ;
- il comprend des moyens d'interface de réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de
- 10 façon que le système embarqué constitue un serveur d'information sur le réseau ; et
- il comprend des moyens d'interface de fichier d'objet, agencés pour établir une correspondance entre des informations transitant par les moyens d'interface de réseau et affectées à au moins ledit fichier d'objet, et des
- 15 informations échangées avec ledit fichier d'objet.
2. Système embarqué selon la revendication 1, dans lequel le fichier d'objet comprend une pièce de logiciel autonome exécutable sur un logiciel de navigation.
3. Système embarqué selon la revendication 1, dans lequel lesdits moyens
- 20 d'interface de réseau sont agencés pour coopérer avec les moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué se comporte comme un client capable de se connecter à au moins un serveur du réseau.
4. Procédé pour instancier un objet situé sur un réseau, caractérisé en ce
- 25 qu'il utilise un système embarqué, équipé d'une puce comprenant des moyens

de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, le système embarqué stockant au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance
5 de cet objet, et comprenant d'une part des moyens d'interface de réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué constitue un serveur d'information sur le réseau, et d'autre part des moyens d'interface de fichier d'objet, agencés pour établir une correspondance entre des informations
10 transitant par les moyens d'interface de réseau et affectées à au moins ledit fichier d'objet, et des informations échangées avec ledit fichier d'objet, le procédé étant en outre caractérisé en ce qu'il permet de décrire un ensemble de sessions entre agents par un fichier d'objet, au moyen d'au moins les étapes suivantes :

15 -établir une liste des agents mis en œuvre ;

-pour chaque agent, définir des arguments d'appel, nécessaires à l'agent.

5. Procédé selon la revendication 4, dans lequel un argument d'appel décrit l'ouverture d'une session avec un autre agent.

6. Procédé selon la revendication 4, dans lequel un agent modifie la liste des
20 arguments utilisés par un autre agent.

7. Procédé pour instancier un objet situé sur un réseau, caractérisé en ce qu'il utilise un système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, le système
25 embarqué stockant au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance de cet objet, et comprenant d'une part des moyens d'interface de réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué constitue un serveur
30 d'information sur le réseau, et d'autre part des moyens d'interface de fichier

d'objet, agencés pour établir une correspondance entre des informations transitant par les moyens d'interface de réseau et affectées à au moins ledit fichier d'objet, et des informations échangées avec ledit fichier d'objet, le procédé étant en outre caractérisé en ce qu'il met en œuvre des sessions entre
5 agents décrites par un fichier d'objet exécuté depuis le serveur d'information du système embarqué au moyen d'au moins les étapes suivantes :

-identification d'un fichier d'objet ;

-exécution de ce fichier d'objet.

8. Procédé selon la revendication 7, dans lequel l'exécution est effectuée par
10 instantiation du premier agent associé au fichier objet.

9. Procédé selon la revendication 7, dans lequel l'exécution est effectuée par instantiation d'un ou plusieurs agents référencés par le fichier objet.

10. Procédé pour instancier un objet situé sur un réseau, caractérisé en ce qu'il utilise un système embarqué, équipé d'une puce comprenant des moyens
15 de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, le système embarqué stockant au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance de cet objet, et comprenant d'une part des moyens d'interface de réseau,
20 agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué constitue un serveur d'information sur le réseau, et d'autre part des moyens d'interface de fichier d'objet, agencés pour établir une correspondance entre des informations transitant par les moyens d'interface de réseau et affectées à au moins ledit
25 fichier d'objet, et des informations échangées avec ledit fichier d'objet, le procédé étant en outre caractérisé en ce qu'il met en œuvre des sessions entre agents décrites par un fichier d'objet exécuté depuis un logiciel de navigation au moyen d'au moins les étapes suivantes :

-chargement par le logiciel de navigation d'un fichier d'objet et d'un logiciel spécifique capable de le mettre en œuvre ;

-exécution du logiciel spécifique par le logiciel de navigation.

11. Procédé selon la revendication 10, dans lequel la réalisation du logiciel
5 spécifique est effectuée au moyen de tout langage interprété, exécutable par le logiciel de navigation.

12. Procédé selon la revendication 10, dans lequel l'interpréteur de fichier objet est réalisé sur un logiciel de navigation.

13. Procédé pour instancier un objet situé sur un réseau, caractérisé en ce
10 qu'il utilise un système embarqué, équipé d'une puce comprenant des moyens de traitement d'information et des moyens de mémorisation d'information, et destiné à coopérer avec un réseau au travers d'un terminal, le système embarqué stockant au moins un fichier d'objet contenant des informations associées à un objet situé sur le réseau et permettant de réaliser une instance
15 de cet objet, et comprenant d'une part des moyens d'interface de réseau, agencés pour coopérer avec des moyens d'interface de réseau appariés, situés dans le terminal, de façon que le système embarqué constitue un serveur d'information sur le réseau, et d'autre part des moyens d'interface de fichier d'objet, agencés pour établir une correspondance entre des informations
20 transitant par les moyens d'interface de réseau et affectées à au moins ledit fichier d'objet, et des informations échangées avec ledit fichier d'objet, le procédé étant en outre caractérisé en ce qu'il permet au système embarqué de rendre possible la mise en œuvre de sessions entre agents décrites par un fichier d'objet exécuté depuis un logiciel de navigation, et en ce qu'il comprend
25 l'étape consistant à identifier, au moyen d'un identificateur universel de ressource, un logiciel spécifique agencé pour mettre en œuvre le logiciel de navigation.

14. Procédé selon la revendication 13, dans lequel l'identificateur universel de ressource est intégré à un document hyper-texte.
15. Procédé selon la revendication 13, dans lequel ledit logiciel spécifique est chargé par une méthode disponible sur le logiciel de navigation et déduite de
- 5 l'identificateur universel de ressource.

1/8

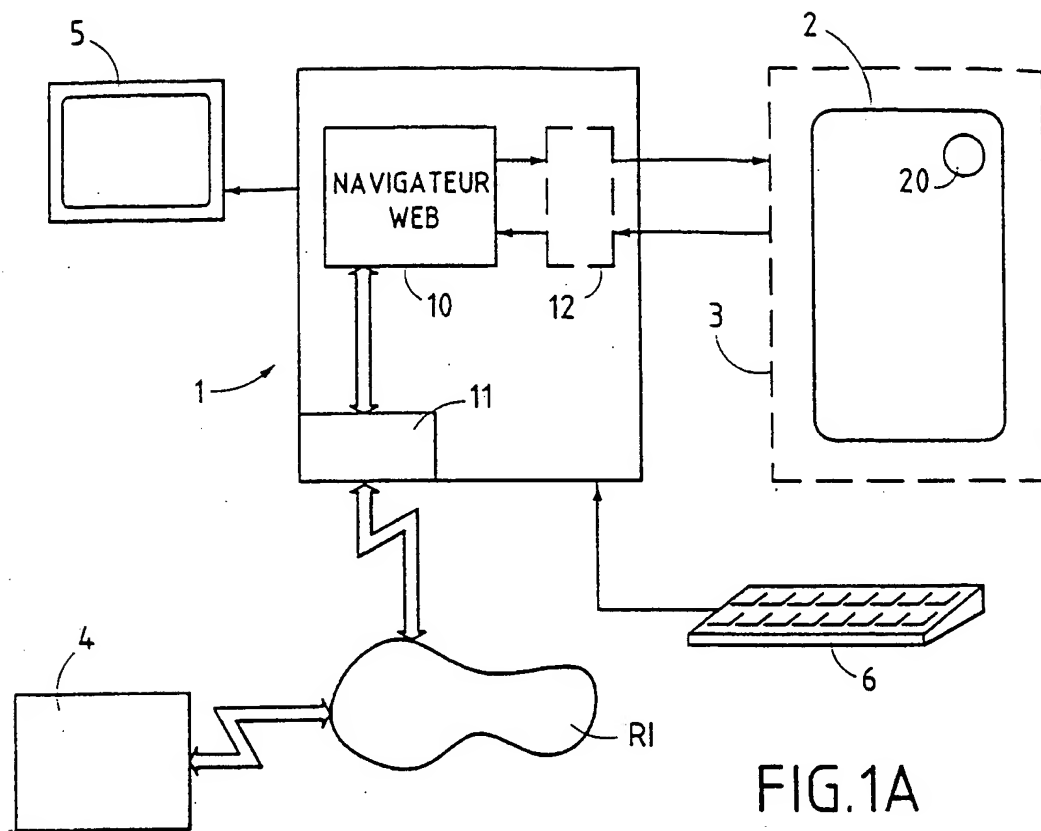


FIG. 1A

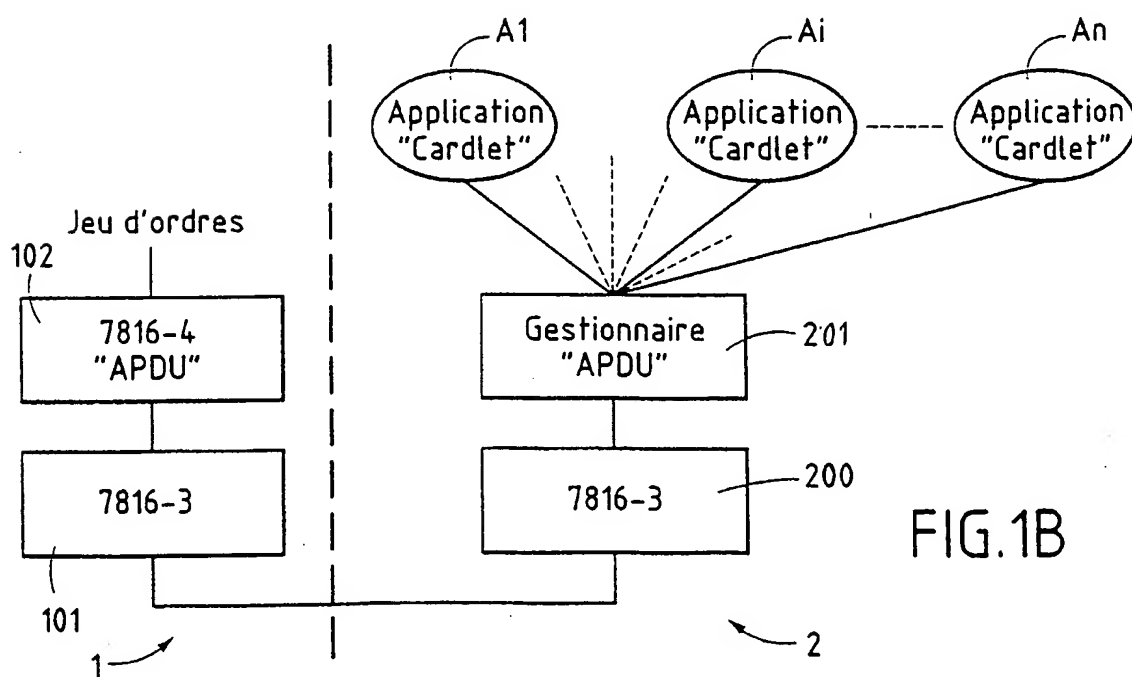


FIG. 1B

FIG. 2

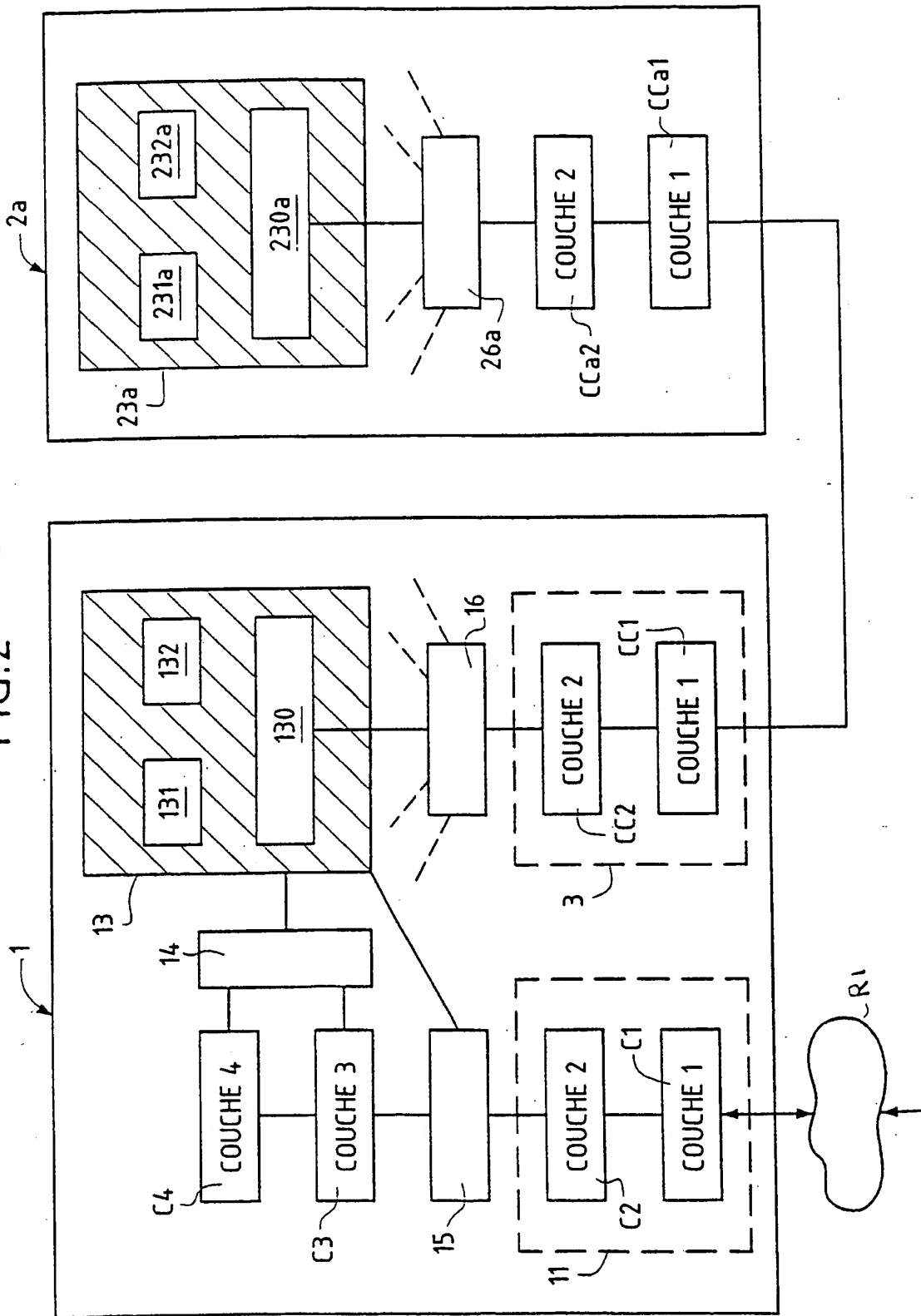


FIG.3

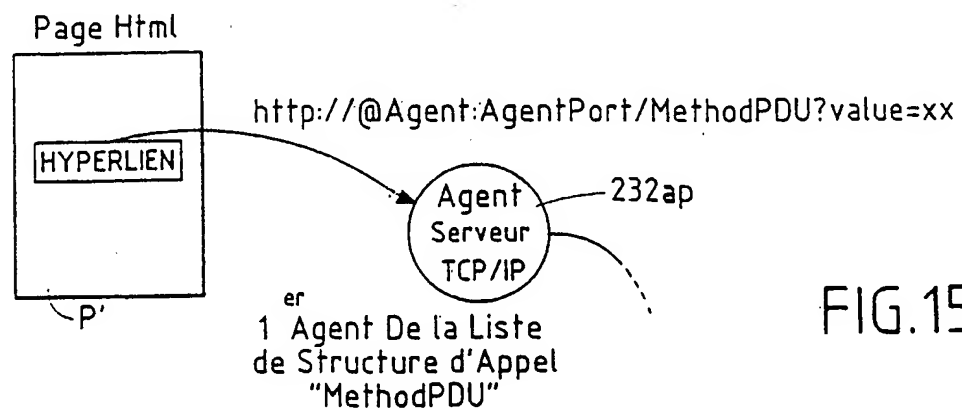
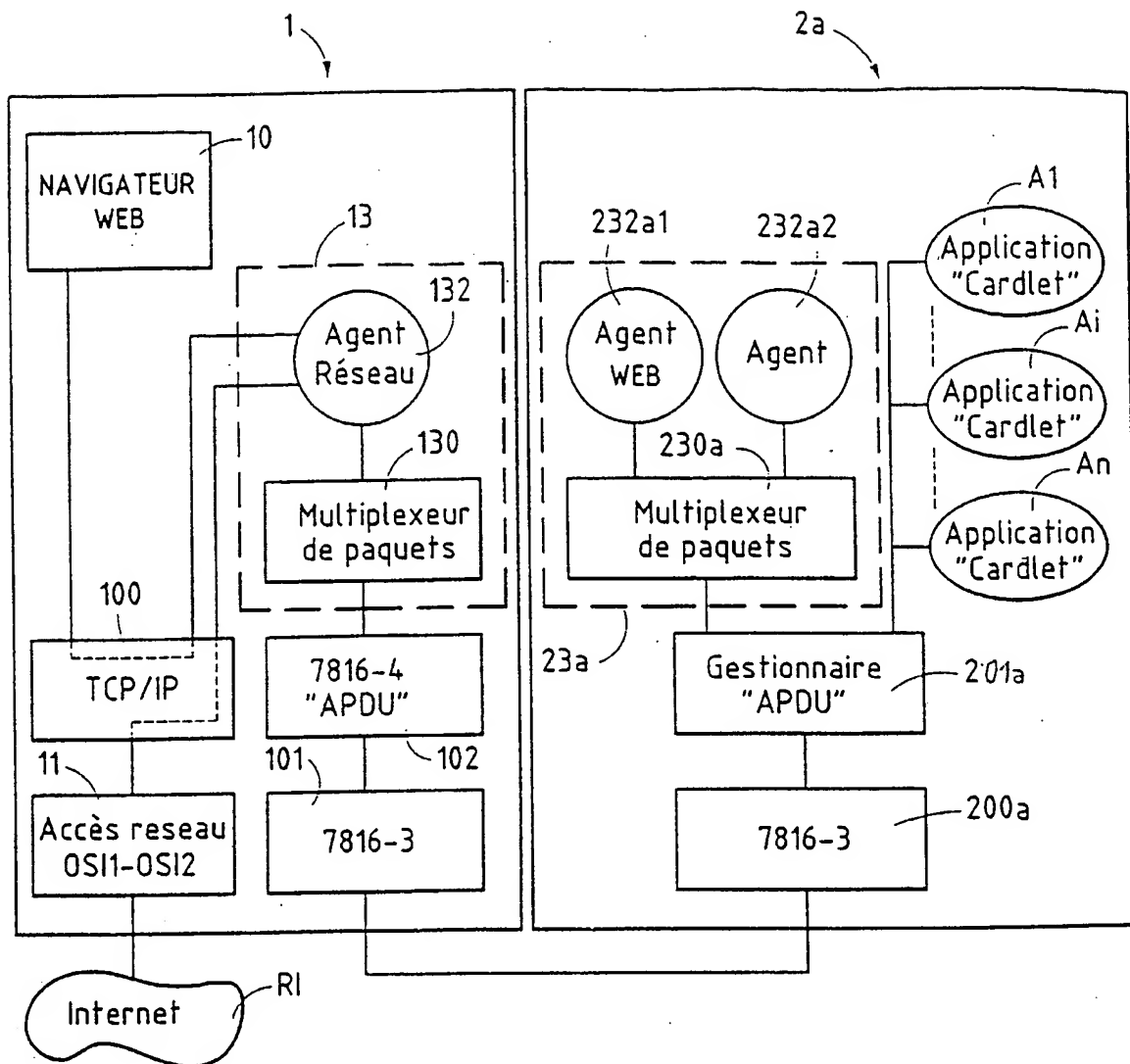


FIG.15

4/8

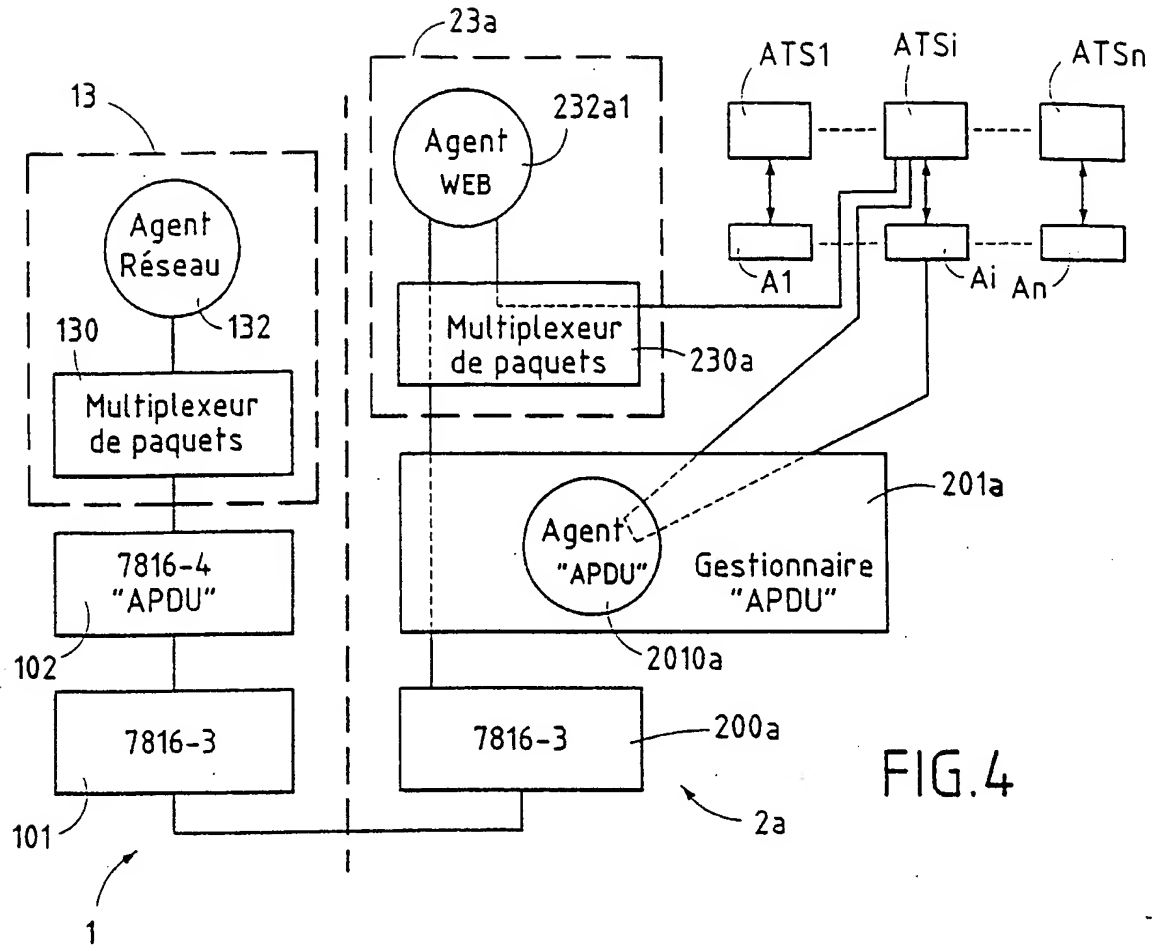


FIG. 4

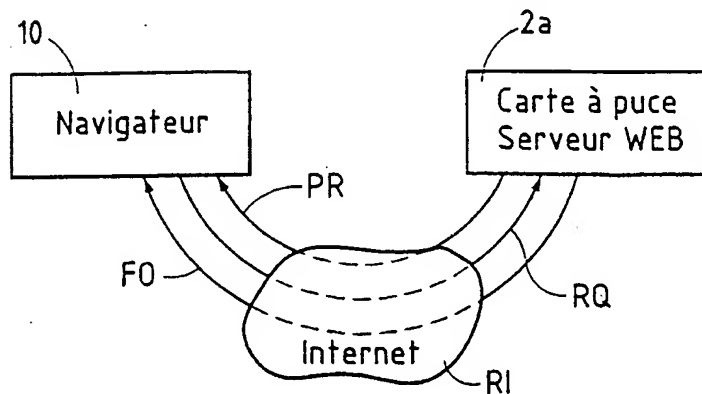


FIG. 5

5/8

FIG.6

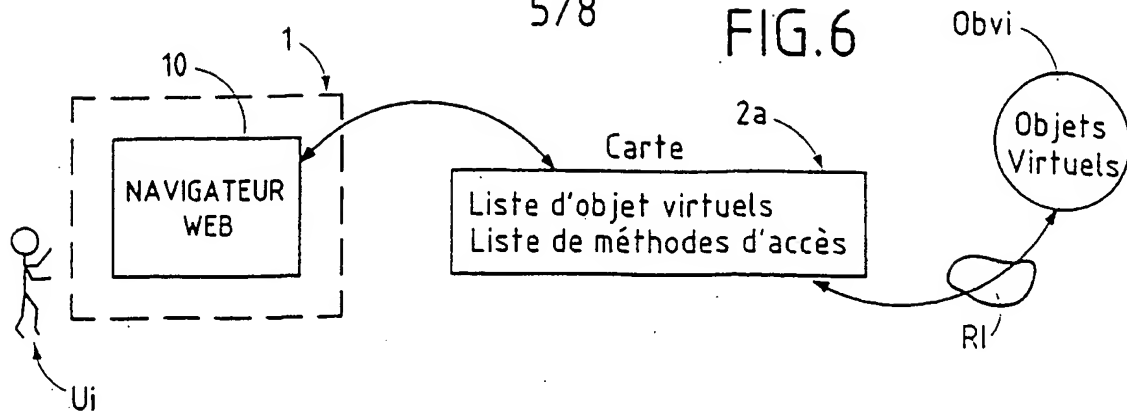


FIG.7

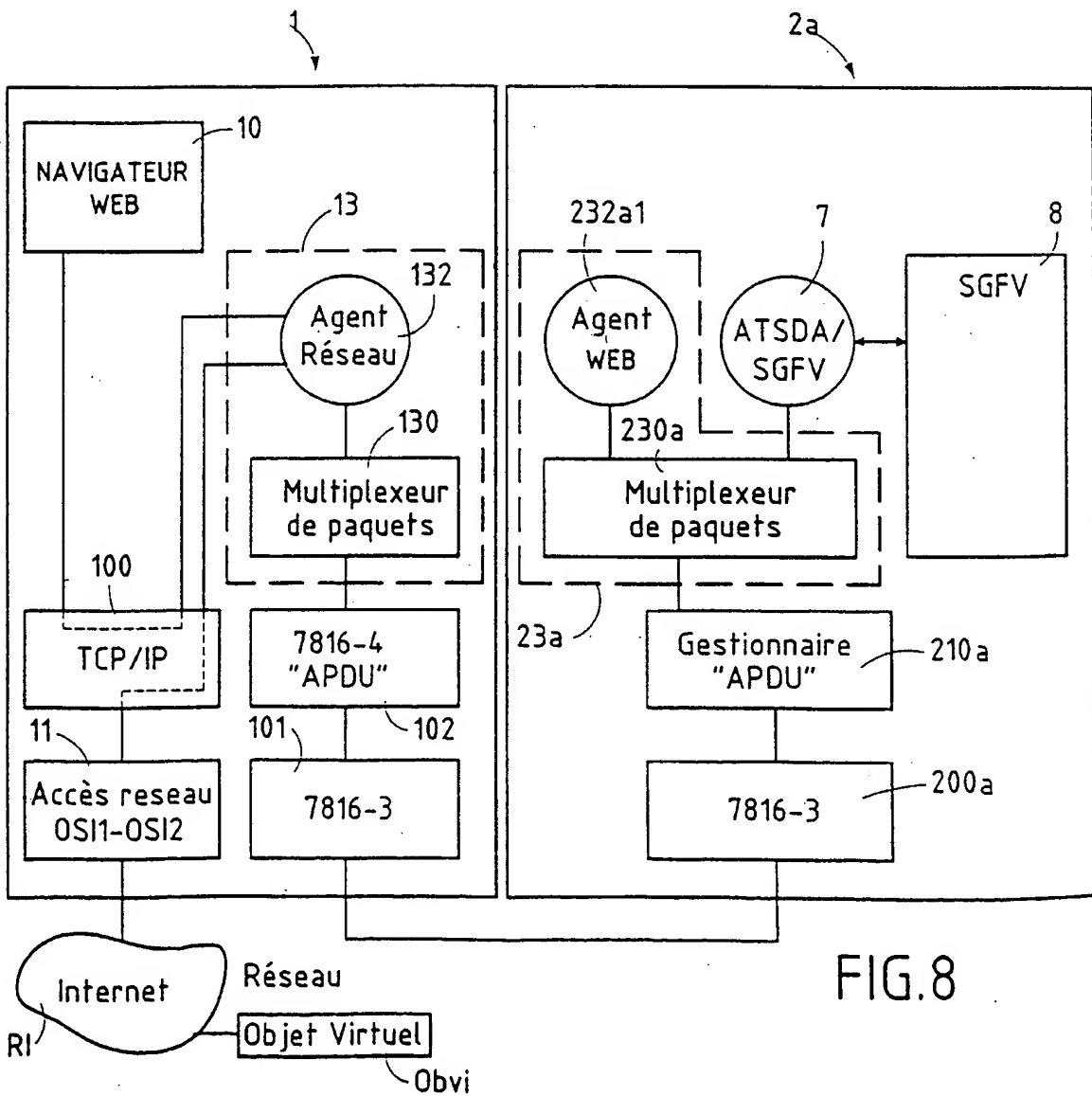
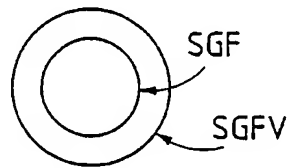


FIG.8

6/8

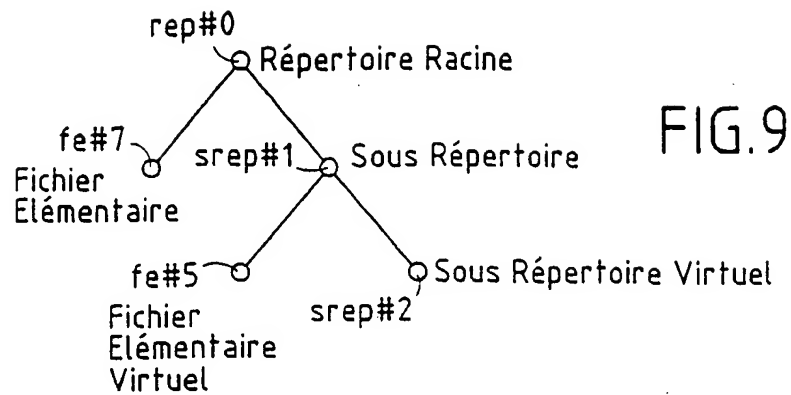


FIG. 9

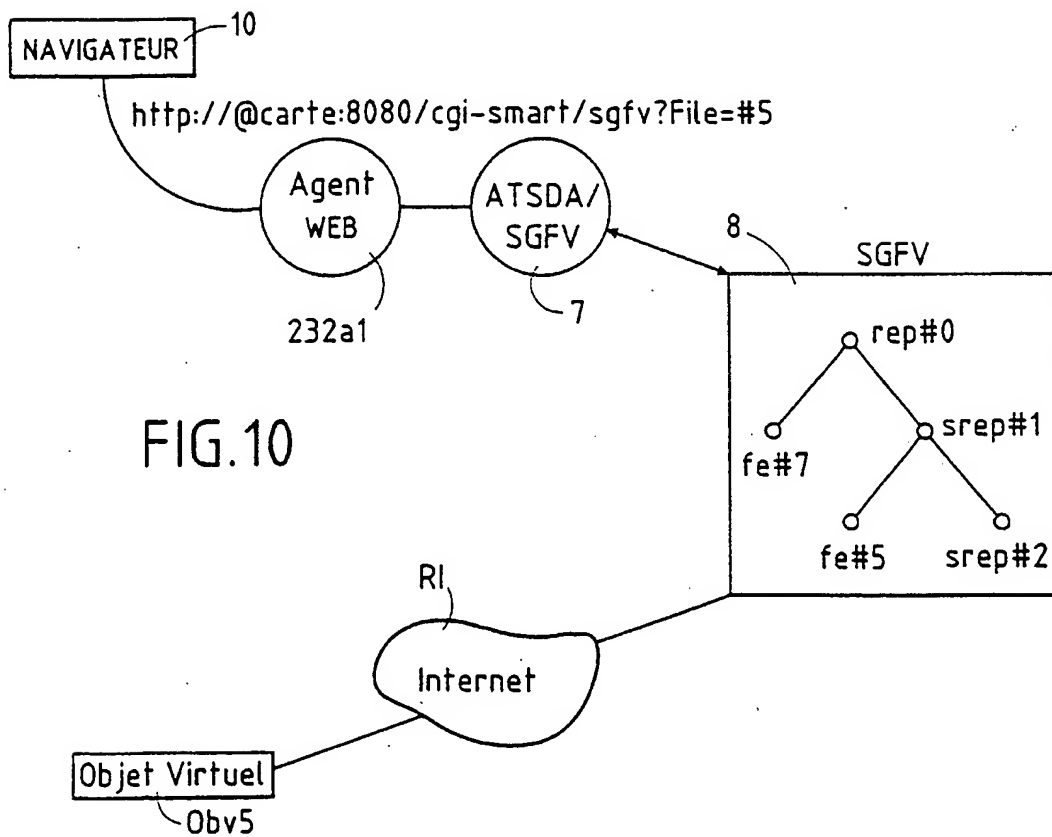


FIG. 10

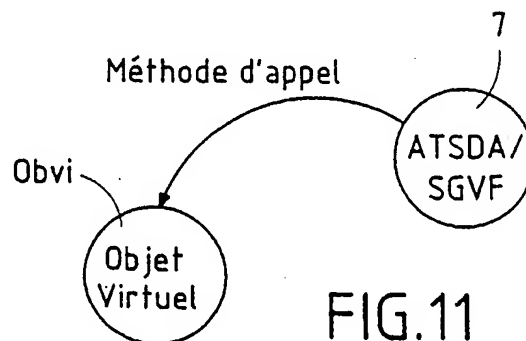
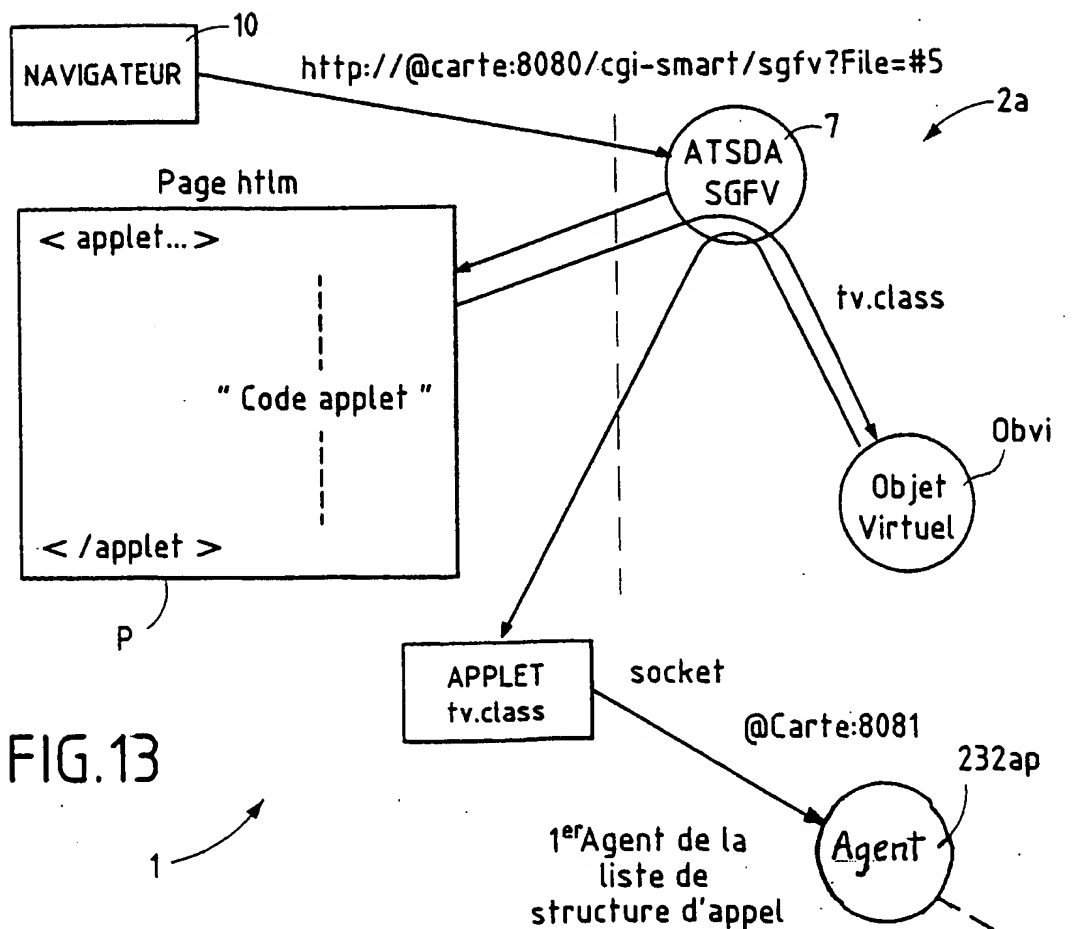
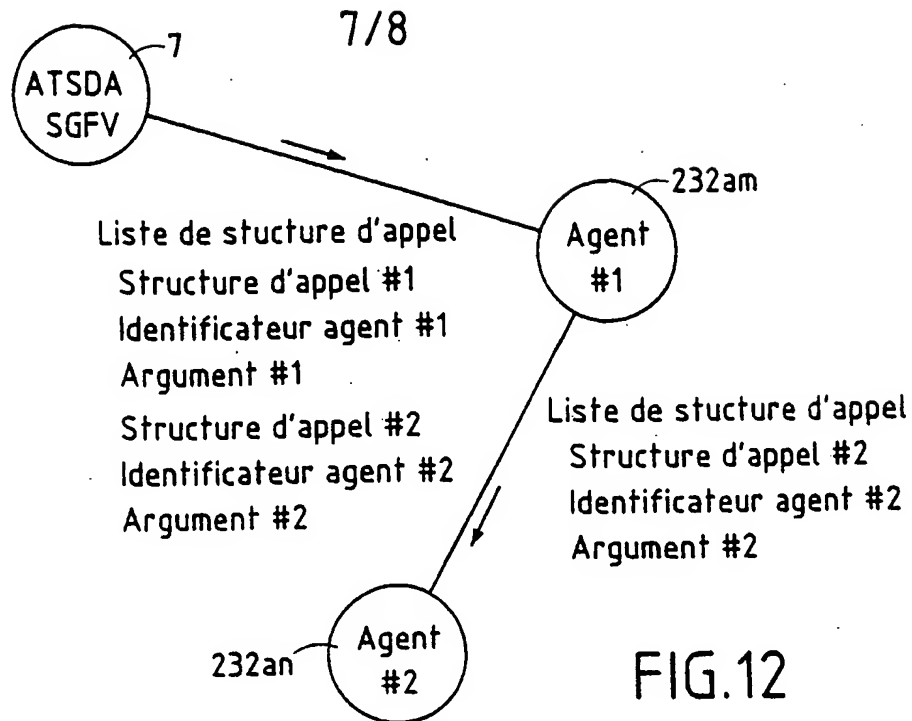


FIG. 11



8/8

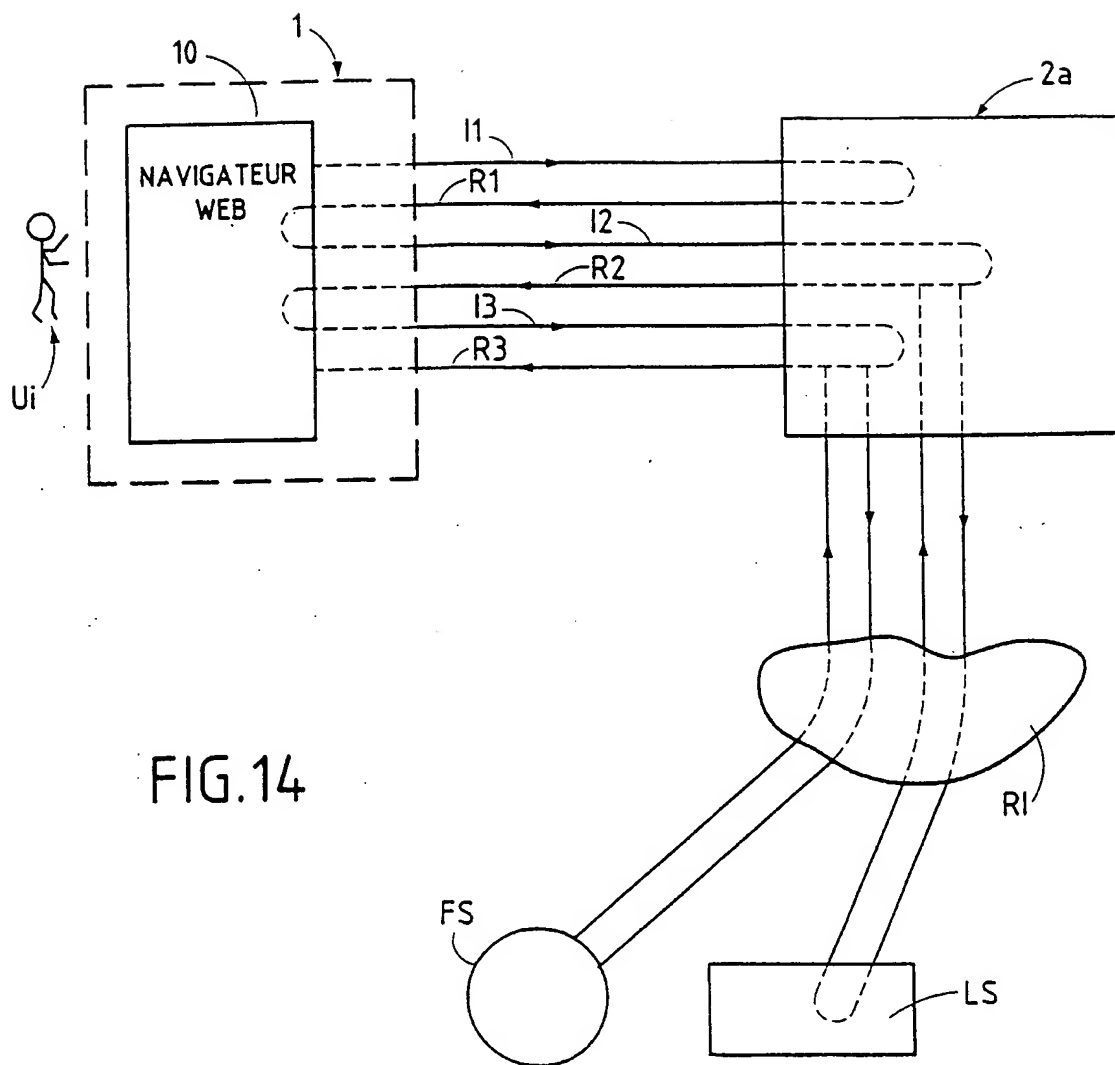


FIG.14

INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR 00/00625

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	WO 98 57474 A (GEMPLUS CARD INT ; MARTINEAU PHILIPPE (FR); MERRIEN LIONEL (US); SI) 17 December 1998 (1998-12-17) page 6, line 5 -page 10, line 9 page 11, line 24 -page 14, line 25 page 15, line 24 -page 19, line 8 page 21, line 15 -page 23, line 4 claims 1-8 figures 1,2 ---	1-3, 7-10, 12-15 4-6, 11
P, X	WO 99 14678 A (WEBTV NETWORKS INC) 25 March 1999 (1999-03-25) page 2, line 9-27 page 5, line 5 -page 7, line 2 page 7, line 24 -page 10, line 16 page 11, line 23 -page 12, line 6 page 12, line 27 -page 16, line 3 ---	1-3, 7, 10, 13-15
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

4 July 2000

Date of mailing of the international search report

14/07/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Lázaro López, M.L.

INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR 00/00625

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	US 5 898 838 A (WAGNER RICHARD HIERS) 27 April 1999 (1999-04-27) column 5, line 54 -column 8, line 4 column 9, line 63 -column 12, line 14 column 14, line 10 -column 15, line 25 column 17, line 59 -column 19, line 8 -----	1-3,7,8, 10-15
A	PEISEL B: "DESIGNING THE NEXT STEP IN INTERNET APPLIANCES" ELECTRONIC DESIGN, US, PENTON PUBLISHING, CLEVELAND, OH, vol. 46, no. 7, 23 March 1998 (1998-03-23), page 50,52,56 XP000780455 ISSN: 0013-4872 page 52, right-hand column, line 20-31 page 56, left-hand column, line 30-56 page 56, right-hand column, line 52 -----	1-15
A	BIGET P ET AL: "How smart cards can benefit from object-oriented technologies" FUTURE GENERATIONS COMPUTER SYSTEMS, NL, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, vol. 13, no. 1, 1 July 1997 (1997-07-01), pages 75-90, XP004081711 ISSN: 0167-739X page 78, left-hand column, line 14 -page 79, left-hand column, line 2 page 79, right-hand column, line 17 -page 80, left-hand column, line 2 page 80, right-hand column, line 12 -page 81, left-hand column, line 14 page 82, left-hand column, line 1-15 page 82, right-hand column, line 18-37 page 85, left-hand column, line 8-19 page 86, left-hand column, line 1-10 page 86, right-hand column, line 1-19 -----	1-15
A	PARADINAS P N VANDEWALE J-J: "A Personal and portable database server: the CQL card" APPLICATIONS OF DATABASES. INTERNATIONAL CONFERENCE, XX, XX, 21 June 1994 (1994-06-21), pages 444-457-457, XP002129612 page 447, line 26-39 page 448, line 12 -page 449, line 35 -----	1-15

INTERNATIONAL SEARCH REPORT

information on patent family members

In International Application No

PCT/FR 00/00625

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9857474 A	17-12-1998	AU 8113798 A ZA 9805151 A	30-12-1998 13-04-1999
WO 9914678 A	25-03-1999	US 5983273 A AU 9127198 A	09-11-1999 05-04-1999
US 5898838 A	27-04-1999	US 5742845 A US 5905908 A	21-04-1998 18-05-1999

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No

PCT/FR 00/00625

A. CLASSEMENT DE L'OBJET DE LA DEMANDE

CIB 7 H04L29/06

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)

CIB 7 H04L

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X A	WO 98 57474 A (GEMPLUS CARD INT ; MARTINEAU PHILIPPE (FR); MERRIEN LIONEL (US); SI) 17 décembre 1998 (1998-12-17) page 6, ligne 5 -page 10, ligne 9 page 11, ligne 24 -page 14, ligne 25 page 15, ligne 24 -page 19, ligne 8 page 21, ligne 15 -page 23, ligne 4 revendications 1-8 figures 1,2	1-3, 7-10, 12-15 4-6,11
P,X	WO 99 14678 A (WEBTV NETWORKS INC) 25 mars 1999 (1999-03-25) page 2, ligne 9-27 page 5, ligne 5 -page 7, ligne 2 page 7, ligne 24 -page 10, ligne 16 page 11, ligne 23 -page 12, ligne 6 page 12, ligne 27 -page 16, ligne 3	1-3,7, 10,13-15
	-/--	

☒ Voir la suite du cadre C pour la fin de la liste des documents

☒ Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

- "A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent
- "E" document antérieur, mais publié à la date de dépôt international ou après cette date
- "L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)
- "O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens
- "P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

- "T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention
- "X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
- "Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
- "&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

4 juillet 2000

Date d'expédition du présent rapport de recherche internationale

14/07/2000

Nom et adresse postale de l'administration chargée de la recherche internationale

Office Européen des Brevets, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 eponl,
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Lázaro López, M.L.

RAPPORT DE RECHERCHE INTERNATIONALE

Dem. le Internationale No

PCT/FR 00/00625

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
P,X	US 5 898 838 A (WAGNER RICHARD HIERS) 27 avril 1999 (1999-04-27) colonne 5, ligne 54 -colonne 8, ligne 4 colonne 9, ligne 63 -colonne 12, ligne 14 colonne 14, ligne 10 -colonne 15, ligne 25 colonne 17, ligne 59 -colonne 19, ligne 8 ---	1-3,7,8, 10-15
A	PEISEL B: "DESIGNING THE NEXT STEP IN INTERNET APPLIANCES" ELECTRONIC DESIGN,US,PENTON PUBLISHING, CLEVELAND, OH, vol. 46, no. 7, 23 mars 1998 (1998-03-23), page 50,52,56 XP000780455 ISSN: 0013-4872 page 52, colonne de droite, ligne 20-31 page 56, colonne de gauche, ligne 30-56 page 56, colonne de droite, ligne 52 ---	1-15
A	BIGET P ET AL: "How smart cards can benefit from object-oriented technologies" FUTURE GENERATIONS COMPUTER SYSTEMS,NL,ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, vol. 13, no. 1, 1 juillet 1997 (1997-07-01), pages 75-90, XP004081711 ISSN: 0167-739X page 78, colonne de gauche, ligne 14 -page 79, colonne de gauche, ligne 2 page 79, colonne de droite, ligne 17 -page 80, colonne de gauche, ligne 2 page 80, colonne de droite, ligne 12 -page 81, colonne de gauche, ligne 14 page 82, colonne de gauche, ligne 1-15 page 82, colonne de droite, ligne 18-37 page 85, colonne de gauche, ligne 8-19 page 86, colonne de gauche, ligne 1-10 page 86, colonne de droite, ligne 1-19 ---	1-15
A	PARADINAS P N VANDEWALE J-J: "A Personal and portable database server: the CQL card" APPLICATIONS OF DATABASES. INTERNATIONAL CONFERENCE,XX,XX, 21 juin 1994 (1994-06-21), pages 444-457-457, XP002129612 page 447, ligne 26-39 page 448, ligne 12 -page 449, ligne 35 -----	1-15

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale No

PCT/FR 00/00625

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 9857474 A	17-12-1998	AU 8113798 A ZA 9805151 A	30-12-1998 13-04-1999
WO 9914678 A	25-03-1999	US 5983273 A AU 9127198 A	09-11-1999 05-04-1999
US 5898838 A	27-04-1999	US 5742845 A US 5905908 A	21-04-1998 18-05-1999